Blindness and Verification of Quantum Computation with One Pure Qubit

Blindness and Verification of Quantum Computation with One Pure Qubit

Theodoros Kapourniotis[1], Elham Kashefi[1], and Animesh Datta[2]

1 **School of Informatics, University of Edinburgh**
10 Crichton Street, Edinburgh EH8 9AB, UK
T.Kapourniotis@sms.ed.ac.uk, ekashefi@inf.ed.ac.uk

2 **Clarendon Laboratory, Department of Physics,**
University of Oxford, OX1 3PU, United Kingdom
animesh.datta@physics.ox.ac.uk

Abstract

While building a universal quantum computer remains challenging, devices of restricted power such as the so-called *one pure qubit* model have attracted considerable attention. An important step in the construction of these limited quantum computational devices is the understanding of whether the verification of the computation within these models could be also performed in the restricted scheme. Encoding via blindness (a cryptographic protocol for delegated computing) has proven successful for the verification of universal quantum computation with a restricted verifier. In this paper, we present the adaptation of this approach to the one pure qubit model, and present the first feasible scheme for the verification of delegated one pure qubit model of quantum computing.

1998 ACM Subject Classification Quantum computation theory, Cryptography, Model verification and validation

Keywords and phrases Delegated Computing, Verification, Measurement-based Model

Digital Object Identifier 10.4230/LIPIcs.TQC.2014.176

1 Introduction

The physical realisation of quantum information processing requires the fulfilment of the five criteria collated by DiVincenzo [13]. While enormous progress had been made in realising them since, we are still some way from constructing a universal quantum computer. This raises the question whether quantum advantages in computation are possible without fulfilling one or more of DiVincenzo's criteria. From a more foundational perspective, the computational power of the intermediate models of computation are of great value and interest in understanding the computational complexity of physical systems. Several such models are known, including fermionic quantum computation [6], instantaneous quantum computation [7], permutational quantum computation [21], and boson sampling [1].

Deeply entwined with the construction of a quantum information processor is the issue of its verification. How do we convince ourselves that the output of a certain computation is correct and obtained using quantum-enhanced means. Depending on a given computation, one or both may be non-trivial. For instance, the correctness of the output of Shor's factoring algorithm [33] can be checked efficiently on a classical machine, but in general this is not known to be possible for all problems solvable by a quantum computer. On the other hand, by allowing a small degree of quantumness to the verifier [2, 18], or considering entangled non-commuting provers [17], the verification problem has been solved for universal quantum


9th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC'14).
Editors: Steven T. Flammia and Aram W. Harrow; pp. 176–204

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

computation. However, not much attention has been given to verifying restricted models of quantum-enhanced computation. It is in this direction that we endeavour to embark.

One of the earliest restricted models of quantum computation was proposed by Knill and Laflamme, named 'Deterministic Quantum Computation with One quantum bit (DQC1)', also referred to as the one pure qubit model [22]. It addresses the challenge of DiVincenzo's first criterion, that of preparing a pure quantum input state, usually the state of n separate qubits in the computational basis state zero. Instead, in the DQC1 model, only one qubit is prepared in a pure state (computational basis zero state) and the rest of the input qubits exist in the maximally mixed state. This model corresponds to a noisier, more feasible experimental setting and was initially motivated by liquid-state NMR proposals for quantum computing. The DQC1 model was shown to be capable of estimating the coefficients of the Pauli operator expansion efficiently. Following this, Shepherd defined the complexity class 'Bounded-error Quantum 1-pure-qubit Polynomial-time (BQ1P)', to capture the power of the DQC1 model [32], and proved that a special case of Pauli operator expansion, the problem of estimating the normalised trace of a unitary matrix to be complete for this class. This problem, and others that can be reduced to it, such as the estimation of the value of the Jones polynomial (see Ref. [12] for more such connections), is interesting from a complexity theoretical point of view since it has no known efficient classical algorithm. Moreover they are not known to belong to the class NP, therefore the problem of verifying the correctness of the result is non-trivial. More recently, it was shown that an ability to simulate classically efficiently a slightly modified version of this model would lead to the collapse of the polynomial hierarchy to the third level [29].

The approach of the Verifiable Universal Blind Quantum Computing (VUBQC) [18] is based on the intermediate step of blind computing, a cryptographic protocol where a restricted client runs the desired computation on a powerful server, such that the server does not learn anything about the delegated computation. A protocol for universal blind quantum computation with a client able to prepare only single qubits, based on Measurement-based Quantum Computing (MBQC) [31] model was introduced in [8]. Here, we take the same approach towards verification by first adapting this existing protocol for blind computing to the DQC1 model. Thus, the first goal is to define what it means to have a DQC1 computation in the MBQC setting. Fixing the input state to almost maximally mixed as it is done in the circuit picture of the DQC1 model does not suffice since the required auxiliary qubits for MBQC could potentially increase the number of pure qubits in the system by more than a logarithmic amount [1]. This adaptation is also necessary as almost all the optimal schemes [2, 8, 15, 25, 4, 27, 28, 34, 23, 19] for the blind computation exploit the possibility of adaptive computation based on the measurement, a freedom not allowed in the original DQC1 model [2]. The main results presented in this paper are the following.

- We introduce a new definition of DQC1 computation within the MBQC framework, called the DQC1-MBQC model [3], which captures the essential property of its original definition in the circuit model. Moreover, we show that the original definition of complexity class BQ1P is contained in DQC1-MBQC, where the latter is able to capture the process where new qubits are introduced or traced out during the execution of the computation.

[1] Increasing the number of pure qubits in the input to the order of logarithmic in the size of the computation is shown not to add extra power to the one pure qubit complexity class [32].
[2] Ref. [26] does not require the server to use measurement-based quantum computing.
[3] We use a different acronym than DQC1 to emphasis the structural distinction with the standard DQC1 model.

- We provide a sufficient condition for a graph state (underlying resource for an MBQC computation [20]) to be usable within DQC1-MBQC. A direct consequence of this is that the universal blind protocol, which satisfies this condition, can be directly adapted to the setting where the server is a DQC1-MBQC machine and the client is able to send one single qubit at a time.
- Building on the blind protocol and adapting the methods presented in [18], a verification protocol for the class DQC1-MBQC with a server restricted to DQC1-MBQC is given, where the probability of the client being forced to accept an incorrect result can be adjusted by setting the security parameter of the model. Since the protocol of [18] does not satisfy the sufficient condition and hence not runnable in the DQC1-MBQC, an alternative method is presented which also leads to different complexity results.

1.1 Preliminaries

We first introduce the notation necessary to describe a computation in MBQC [31, 11]. A generic pattern, consists of a sequence of commands acting on qubits:

- $N_i(|q\rangle)$: Prepare the single auxiliary qubit i in the state $|q\rangle$;
- $E_{i,j}$: Apply entangling operator controlled-Z to qubits i and j;
- M_i^α: Measure qubit i in the basis $\{\frac{1}{\sqrt{2}}(|0\rangle + e^{i\alpha}|1\rangle), \frac{1}{\sqrt{2}}(|0\rangle - e^{i\alpha}|1\rangle)\}$ followed by trace out the measured qubit. The outcome of measuring qubit i is called result and is denoted by s_i;
- $X_i^{s_j}, Z_i^{s_j}$: Apply a Pauli X or Z correction on qubit i depending on the result s_j of the measurement on the j-th qubit.

The corrections could be combined with measurements to perform 'adaptive measurements' denoted as ${}^{s_z}[M_i^\alpha]^{s_x} = M_i^{(-1)^{s_x}\alpha + s_z\pi}$. A pattern is formally defined by the choice of a finite set V of qubits, two not necessarily disjoint sets the input and the output, $I \subset V$ and $O \subset V$ determining the pattern inputs and outputs, and a finite sequence of commands acting on V.

▶ **Definition 1** ([10]). A pattern is said to be runnable if

(R0) no command depends on an outcome not yet measured;

(R1) no command (except the preparation) acts on a measured or not yet prepared qubit;

(R2) a qubit is measured (prepared) if and only if it is not an output (input).

The entangling commands $E_{i,j}$ define an undirected graph over V referred to as (G, I, O). Along with the pattern we define a partial order of measurements and a dependency function d which is a partial function from O^C to $\mathcal{P}^{I^C}$, where $\mathcal{P}$ denotes the power set. Then, $j \in d(i)$ if j gets a correction depending on the measurement outcome of i. In what follows, we will focus on patterns that realise (strongly) deterministic computation, which means that the pattern implements a unitary on the input up to a global phase. A sufficient condition on the geometry of the graph state to allow unitary computation is given in [10, 9] and will be used later in this paper. In what follows, $x \sim y$ denotes that x is adjacent to y in G.

▶ **Definition 2** ([10]). A *flow* $(f, \preceq)$ for a geometry (G, I, O) consists of a map $f : O^c \mapsto I^c$ and a partial order $\preceq$ over V such that for all $x \in O^c$

(F0) $x \sim f(x)$;

(F1) $x \preceq f(x)$;

(F2) for all $x, y : y \neq x, y \sim f(x)$ we have $x \preceq y$.

1.2 Main results

1.2.1 DQC1-MBQC

We define the class BQ1P formally as introduced by Shepherd [32], and then recast it into the MBQC framework.

▶ **Definition 3** (Bounded-error Quantum 1-pure-qubit Polynomial-time complexity class, [32])**.** BQ1P is defined using a bounded-error uniform family of quantum circuits – DQC1. A DQC1 circuit takes as input a classical string $\boldsymbol{x}$, of size n, which encodes a fixed choice of unitary operators applied on a standard input state $|0\rangle\langle 0| \otimes I_{w-1}/2^{w-1}$. The width of the circuit w is polynomially bounded in n. Let $Q_n(x)$ be the result of measuring the first qubit of the final state of a DQC1 circuit. A language in BQ1P is defined by the following rule:

$$\forall a \in L : Pr(Q_n(a) = 1) \geq \frac{1}{2} + \frac{1}{2q(n)} \tag{1}$$

$$\forall a \notin L : Pr(Q_n(a) = 1) \leq \frac{1}{2} - \frac{1}{2q(n)} \tag{2}$$

for some polynomially bounded q(n).

An essential physical property of DQC1 that we mean to preserve in DQC1-MBQC is its limited purity. To capture this we introduce the *purity parameter*:

$$\pi(\rho) = \log_2 \left(\mathrm{Tr}(\rho^2)\right) + d, \tag{3}$$

where d is the logarithm of the dimension of the state ρ. For a DQC1 circuit with k pure qubits, at each state of the computation the value of purity parameter π for that state remains constant equal to k. In fact, Shepherd showed that the class BQ1P is not extended by increasing the number of pure input qubits logarithmically. Thus, a purity that does not scale too rapidly with the problem size still remains in the same complexity class.

A characterisation of MBQC patterns compatible with the idea of the DQC1 model as introduced above is presented next. Any MBQC pattern is called DQC1-MBQC when there exists a runnable rewriting of this pattern such that after every elementary operation (for any possible branching of the pattern) the purity parameter π does not increase over a fixed constant. We assume that the system at the beginning has only the input state and at the end has only the output state.

We define a new complexity class that captures the idea of one pure qubit computation in the MBQC model. This complexity class, that we name DQC1-MBQC, can be based on any universal DQC1-MBQC resource pattern, which is defined analogously to the DQC1 circuits [32] as a pattern that can be adapted to execute any DQC1-MBQC pattern of polynomial size. A particular example of such a resource, as we will present later, can be built using the the brickwork state of [8] designed for the purpose of universal blind quantum computing. The input to a universal pattern is the description of a computation as a measurement angle vector and is used to classically control the measurements of the MBQC pattern. The quantum input of the open graph is always fixed to a mostly maximally mixed state, in correspondence to the DQC1 model.

▶ **Definition 4.** A language in DQC1-MBQC complexity class is defined based on a universal DQC1-MBQC resource pattern P_α that takes as input an angle vector $\boldsymbol{\alpha}$ of size n and is applied on the quantum state $|+\rangle\langle +| \otimes I_{w-1}/2^{w-1}$, $w \in O(n)$. A word $\boldsymbol{\alpha}$ belongs to the

language depending of the probabilities of the measurement outcome ($R_n(\boldsymbol{\alpha})$) of the first output qubit of pattern P_α which are defined identically to Definition 3:

$$\forall a \in L : Pr(R_n(\alpha) = 1) \geq \frac{1}{2} + \frac{1}{2r(n)} \tag{4}$$

$$\forall a \notin L : Pr(R_n(\alpha) = 1) \leq \frac{1}{2} - \frac{1}{2r(n)} \tag{5}$$

for some polynomially bounded r(n).

▶ **Corollary 5.** *BQ1P ⊆ DQC1-MQBC.*

Proof. Any circuit description using a fixed set of gates can be efficiently translated into a measurement pattern applicable on the brickwork state. A specific example of translating each gate from the universal set {Hadamard, $\pi/8$, c-NOT} to a 'brick' element of the brickwork state is given in [8]. The quantum input state in the resulting measurement pattern is in the almost-maximally-mixed state, therefore the pattern is a valid DQC1-MBQC pattern. ◀

▶ **Definition 6.** An MBQC pattern is a DQC1-MBQC pattern if there is a runnable sequence of commands where for every elementary command and measurement outcome, there exists a fixed constant value c such that the overall quantum state of the system (ρ_i with dimension d_i) after the i^{th} operation satisfies the following relation

$$\pi(\rho_i) < \pi(\rho_{in}) + c, \tag{6}$$

where ρ_{in} is the quantum input of the pattern with dimension d_{in}, which is fixed to be the product of c_{in} (constant) pure qubits and a maximally mixed state of $d_{in} - c_{in}$ qubits.

The above definition captures the essence of DQC1 in that it maintains a low purity, high entropy state in MBQC, in contrast to DiVincenzo's first criterion. We derive a sufficient condition (that is also constructive) for the open graph state leading to DQC1-MBQC, capturing the universal blind quantum computing protocol as a special case. However, a general characterisation and further structural link with determinism in MBQC [10, 9, 24] is left as an open question for future work.

▶ **Theorem 7.** *Any measurement pattern on an open graph state (G, I, O) with flow $(f, \preceq)$ (as defined in Definition 2) and measurement angles $\boldsymbol{\alpha}$ where either $|I| = |O|$ or the flow function is surjective and all auxiliary preparations are on the $(X - Y)$ plane represents a DQC1-MBQC pattern.*

The full details and the proof of this theorem is provided in Section 2.

1.2.2 Blindness

A direct consequence of Theorem 7 is that the Universal Blind Computing Protocol (UBQC) introduced in [8] can be easily adapted to fit within the DQC1-MBQC class, since it is based on an MBQC pattern on a graph state with surjective flow.

In the blind cryptographic setting a client (Alice) wants to delegate the execution of an MBQC pattern to a more powerful server (Bob) and hide the information at the same time. The UBQC protocol is based on the separation of the classical and quantum operations when running an MBQC pattern. The client prepares some randomly rotated quantum states and sends them to the server and from this point on the server executes the quantum operations on them (entangling according to the graph and measuring) and the client calculates the

measurement angles for the server and corrects the measurement outcomes she receives (to undo the randomness and get the correct result).

To define blindness formally we allow Bob to deviate from the normal execution in any possible way, and this is captured by modelling his behaviour during the protocol by an arbitrary CPTP map. The main requirement for blindness is that for any input and averaged over all possible choices of parameters by Alice, Bob's final state can always be written as a fixed CPTP map applied on his initial state, thus not offering any new knowledge to him. This definition of stand-alone blindness was presented first in [14] and takes into account the issue of prior knowledge.

▶ **Definition 8** (Blindness)**.** Let P be a protocol for delegated computation: Alice's input is a description of a computation on a quantum input, which she needs to perform with the aid of Bob and return the correct quantum output. Let ρ_{AB} express the joint initial state of Alice and Bob and σ_{AB} their joint final state, when Bob is allowed to do any deviation from the correct operation during the execution of P, averaged over all possible choices of random parameters by Alice. The protocol P is blind iff

$$\forall \rho_{AB} \in \mathcal{L}(\mathcal{H}_{AB}), \exists\ \mathcal{E} : \mathcal{L}(\mathcal{H}_B) \to \mathcal{L}(\mathcal{H}_B), \text{ s.t.} \quad \mathrm{Tr}_A(\sigma_{AB}) = \mathcal{E}(\mathrm{Tr}_A(\rho_{AB})) \tag{7}$$

To adapt the original UBQC protocol into the DQC1-MBQC setting we change the order of the operations so that the client does not send all the qubits to the server at the beginning, but during the execution of the pattern, following a rewriting of the pattern that is consistent with the purity requirement. The details are described in Section 2.

▶ **Theorem 9.** *There exists a blind protocol for any DQC1-MBQC computation where the client is restricted to BPP and the ability to prepare single qubits and the server is within DQC1-MBQC.*

1.2.3 Verification

In the verification cryptographic setting a client (Alice) wants to delegate a quantum computation to a more powerful server (Bob) and accept if the result is correct or reject if the result is incorrect (server is behaving dishonestly). The main idea of the original protocol of [18] is to test Bob's honesty by hiding a trap qubit among the others in the resource state sent to him by Alice. Blindness means that Bob cannot learn the position of the trap, nor its state. During the execution of the pattern Bob is asked to measure this trap qubit and report the result to Alice. If Bob is honest this measurement gives a deterministic result, which can be verified by Alice. Bob being dishonest means that Alice will receive the wrong result with no-zero probability. Depending on that result, Alice accepts or rejects the final output received by Bob.

To define verifiability formally we need to establish an important difference with the original protocol [18]: In a DQC1-MBQC pattern the quantum input is in a mixed state as opposed to a pure state. Reverting to the original definition that derives from the quantum authentication schemes in [3] we need to add an extra reference system R, that is used to purify the mixed input that exists in Alice's private system A. The assumption is that Bob does not learn anything about the reference system (ex. Alice is provided with the quantum input from a third trusted party which also holds the purification). Bob is allowed to choose any possible cheating strategy and our goal is to minimise the probability of Alice accepting the incorrect output of the computation at the end of the protocol.

▶ **Definition 10.** A protocol for delegated computation is ϵ-verifiable ($0 \leq \epsilon < 1$) if for any choice of Bob's strategy j, it holds that for any input of Alice:

$$\mathrm{Tr}(\sum_{\nu} p(\nu) P^{\nu}_{\mathrm{incorrect}} B_j(\nu)) \leq \epsilon \tag{8}$$

where $B_j(\nu)$ is the state of Alice's system A together with the purification system R at the end of the run of the protocol, for choice of Alice's random parameters ν and Bob's strategy j. If Bob is honest we denote this state by $B_0(\nu)$. Let $P_\perp$ be the projection onto the orthogonal complement of the the correct (purified) quantum output. Then,

$$P^{\nu}_{\mathrm{incorrect}} = P_\perp \otimes |\eta_t^{\nu_c}\rangle\langle\eta_t^{\nu_c}| \tag{9}$$

where $|\eta_t^{\nu_c}\rangle$ is a state that indicates if Alice accept or reject the result (see Section 3).

A verification protocol should also be correct, which means that in case Bob is honest Alice's state at the end of the run of the protocol is the correct output of the computation and an extra qubit set in the accept state (this property is also referred to as completeness).

In VUBQC, in order to adjust the parameter ϵ to any arbitrary value between 0 and 1 (a technique called probability amplification), one needs to add polynomially many trap qubits within the MBQC pattern. Specifically, adding polynomially many traps and incorporating the pattern into a fault tolerance scheme that corrects d errors, gives parameter ϵ exponentially small on d. As we explain in Section 3, adding polynomially many traps, following the same scheme as VUBQC, creates a pattern that is not a DQC1-MBQC pattern. Therefore to achieve an amplification of the error probability we need to develop a modified trapping scheme.

In Section 3 we give a verification protocol for DQC1-MBQC problems where, instead of running the pattern once, s computations of the same size are run in series, one being the actual computation and the others being trap computations. A similar approach is also considered for the restricted setting of the photonic implementation of VUBQC [5] and a verification protocol of the entanglement states [30]. In our setting each trap computation contains an isolated trap injected in a random position between the qubits of the pattern. We prove that in this verification protocol the server is within DQC1-MBQC complexity class, while the client is within BPP together with single qubit preparations (as in the original VUBQC). Moreover in this verification protocol we achieve the goal of probability amplification by choosing the appropriate value for parameter s.

▶ **Theorem 11.** *There exists a correct ϵ-verifiable protocol where the client is restricted to BPP and the ability to prepare single qubits and the server is within DQC1-MBQC. Using $O(sm)$ qubits and $O(sm)$ time steps, where m is the size of the input computation, we have:*

$$\epsilon = \frac{2m}{s} \tag{10}$$

2 DQC1-MBQC and Blindness

In this section we give a constructive proof of our main theorem for DQC1-MBQC and show how to construct a blind protocol as a consequence. The first step for proving Theorem 7 is the following rewriting scheme for patterns with flow.

▶ **Lemma 12.** *Any measurement pattern on an open graph state (G, I, O) with flow $(f, \preceq)$ (as defined in Definition 2) and measurement angles $\boldsymbol{a}$ where either $|I| = |O|$ or the flow*

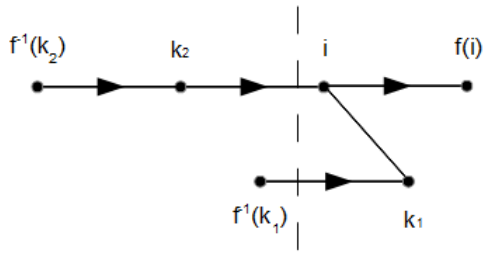

Figure 1 Qubit i gets an X correction from k_2 and Z corrections from $f^{-1}(k_2)$ and $f^{-1}(k_1)$. Qubits on the left of the dashed line are in the past of i. Qubit k_1 is created at timestep $f^{-1}(k_1)$ which is before timestep i from flow condition (F2).

function is surjective can be rewritten as

$$P_{\boldsymbol{a}} = \prod_{i\in O} X_i^{S_i^x} Z_i^{S_i^z} \prod_{i\in O^c}^{\prec} \left({}^{S_i^z}[M_i^{a_i}]^{S_i^x} \left(\prod_{\{k:k\sim i, k\succeq i\}} E_{i,k} \right) N_{f(i)}(|+\rangle) \right) \tag{11}$$

where $S_i^x = s_{f^{-1}(i)}$ *for* $i \in I^c$, *else* $S_i^x = 0$ *and* $S_i^z = \sum_{\{k:k\in I^c, k\sim i, i\neq f^{-1}(k)\}} s_{f^{-1}(k)} \mod 2$. *The above pattern is runnable and implements the following unitary*

$$U_{G,I,O,\boldsymbol{a}} = 2^{|O^c|/2} \left(\prod_{i\in O^c} \langle +_{a_i}|_i \right) E_G N_{I^c} \tag{12}$$

where E_G *and* N_{I^c} *represent the global entangling operator and global preparation respectively.*

Proof. First we need to prove that $P_{\boldsymbol{a}}$ is runnable (cf. Definition 1). For condition (R0) we make the following observations: At step i, for $i \in I^c$, we need the result $s_{f^{-1}(i)}$ which is generated at step $f^{-1}(i)$, where $f^{-1}(i) \prec i$ from flow condition (F1). We also need the results $s_{f^{-1}(k)}$, for $\{k : k \in I^c, k \sim i, i \neq f^{-1}(k)\}$, which are generated at step $f^{-1}(k)$, where $f^{-1}(k) \prec i$ from flow condition (F2). Thus, condition (R0) is satisfied (see Figure 1 for a particular example). For condition (R1) we make the following observations: At step i, for $i \in O^c$, the entangling operator and measurement operator act on qubit i which either belongs in the set of inputs I or is created at step $f^{-1}(i)$, where $f^{-1}(i) \prec i$ from flow condition (F1). Entangling operator acts also on qubits $\{k : k \sim i, k \succeq i\}$. If $k = f(i)$ then qubit k is created at the same step (i) by operator $N_{f(i)}$. If $k \neq f(i)$ then qubit k is either an input or it is created at step $f^{-1}(k)$, and we have by flow condition (F2): i is a neighbour of k and $i \neq f^{-1}(k)$, thus $f^{-1}(k) \prec i$ (Figure 1). Final correction operators act on qubits that belong to the set of outputs O, which either belong also to the set of inputs I or are created at steps $\{f^{-1}(i) : i \in O\}$, where $\forall i \in O \setminus I, f^{-1}(i) \prec i$ from flow condition (F1). Moreover they have not yet been measured since $i \notin O^C$. Thus, condition (R1) is satisfied. It is easy to see that condition (R2) is satisfied.

Next we prove that the pattern of Equation 11 is implementing the unitary operation of Equation 12 when applied on an open graph with the properties described above. Since condition (R1) is satisfied, all preparation operators trivially commute with all previous operators

$$P_{\boldsymbol{a}} = \prod_{i\in O} X_i^{S_i^x} Z_i^{S_i^z} \prod_{i\in O^c}^{\prec} \left({}^{S_i^z}[M_i^{a_i}]^{S_i^x} \left(\prod_{\{k:k\sim i, k\succeq i\}} E_{i,k} \right) \right) N_{I^c}\,.$$

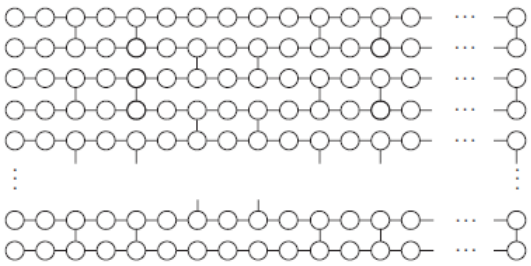

Figure 2 Brickwork state.

Each entangling operator commutes with all previous measurements since it is applied on qubits with indices $\succeq i$.

$$P_{\boldsymbol{a}} = \prod_{i\in O} X_i^{S_i^x} Z_i^{S_i^z} \prod_{i\in O^c}^{\preceq} \left({}^{S_i^z}[M_i^{a_i}]^{S_i^x} \right) E_G N_{I^c} .$$

We can decompose the conditional measurements into simple measurements and corrections

$$P_{\boldsymbol{a}} = \prod_{i\in O} X_i^{S_i^x} Z_i^{S_i^z} \prod_{i\in O^c}^{\preceq} \left(M_i^{a_i} X_i^{S_i^x} Z_i^{S_i^z} \right) E_G N_{I^c} .$$

By rearranging the order of correction operators we take

$$P_{\boldsymbol{a}} = \prod_{i\in O^c}^{\preceq} \left(X_{f(i)}^{s_i} \prod_{\{k:k\sim f(i),k\neq i\}} Z_k^{s_i} M_i^{a_i} \right) E_G N_{I^c} .$$

The above equation implements the unitary operation presented in the lemma (Equation 12) as proved in [10]. ◀

Next, we notice that there exist many universal families of open graph states satisfying the requirements of the above lemma. One such example is the brickwork graph state originally defined in [8]. In this graph state (Figure 2), the subset of vertices of the first column correspond to the input qubits I and the subset of vertices of the final column correspond to the output qubits O. This graph state has flow function $f((i,j)) = (i, j+1)$ and the following partial order for measuring the qubits: $\{(1,1),(2,1),\ldots,(w,1)\} \prec \{(1,2),(2,2),\ldots,(w,2)\} \prec \ldots \prec \{(1,d-1),(2,d-1),\ldots,(w,d-1)\}$, where w is the width and d is the depth of the graph and hence from Lemma 12 we obtain the following corollary.

▶ **Corollary 13.** *Any computation over the brickwork open graph state G with qubit index $(i \leq w, j \leq d)$ can be rewritten as follows.*

$$P_{\boldsymbol{a}} = \prod_{i=1}^{w} X_{(i,d)}^{S_{(i,d)}^x} Z_{(i,d)}^{S_{(i,d)}^z} \prod_{j=1}^{d-1} \prod_{i=1}^{w} {}^{S_{(i,j)}^z}\left[M_{(i,j)}^{a_{(i,j)}}\right]^{S_{(i,j)}^x} \left(\prod_{\substack{\{k,l:(k,l)\sim(i,j),\\ k\geq i, l\geq j\}}} E_{(i,j),(k,l)} \right) N_{(i,j+1)} \quad (13)$$

where

$$S_{(i,j)}^x = s_{(i,j-1)} \quad \textit{for } j>1, \textit{ else } S_{(i,1)}^x = 0, \textit{ and}$$

$$S_{(i,j)}^z = \sum_{\{k,l:(k,l)\sim(i,j),l\leq j\}} s_{(k,l-1)} \mod 2 \textit{ for } j>2, \textit{ else } S_{(i,j)}^z = 0.$$

We show that patterns defined in Lemma 12 are within the framework of Definition 6 hence obtaining a sufficient condition for DQC1-MBQC.

▶ **Theorem 1.** *Any measurement pattern that can be rewritten in the form of Equation 11 represents a DQC1-MBQC pattern.*

Proof. A first general observation about the purity parameter π is that adding a new pure qubit σ to state ρ means that π increases by unity

$$\pi_{\rho\otimes\sigma} = \log_2 \mathrm{Tr}((\rho\otimes\sigma)^2) + d + 1 = \log_2 \mathrm{Tr}(\rho^2)\mathrm{Tr}(\sigma^2) + d + 1 = \pi_\rho + 1.$$

Additionally, applying any unitary U does not change the purity parameter π of the system since $\mathrm{Tr}((U\rho U^\dagger)^2) = \mathrm{Tr}(\rho^2)$ and dimension remains the same.

Returning to Equation 11, we notice that for every step $i \in O^c$ of the product the total computation performed corresponds mathematically to the following: On the qubit tagged with position i, a $J(a'_i)$ unitary gate is applied (where a'_i is an angle that depends on a_i and previous measurement results) up to a specific Pauli correction (depending on the known measurement result) and some specific Pauli corrections on the its entangled neighbours (again depending on the measurement result). At the end the qubit is tagged with position $f(i)$ (where f is the flow function). Since this mathematically equivalent computation is a unitary and the dimension of the system remains the same (there is only a change of position tags) we conclude that each step $i \in O^c$ does not increase the purity parameter of the system. To finish the proof we need to ensure that the individual operations within each step $i \in O^c$ and for $i \in O$ do not increase the purity parameter by more than a constant (and since there is only a constant number of operations within each step this does not increase the purity at any point more than constant). This is true since all these operations apply on (or add or trace over) a constant number of qubits. ◀

Building on this result, we can translate the UBQC protocol of [8] (and in fact many other existing protocols) to allow the blind execution of any DQC1-MBQC computation, where the server is restricted to DQC1-MBQC complexity class. The UBQC protocol is based on the brickwork graph state described above. Alice prepares all the qubits of the graph state, adding a random rotation around the (X, Y) plane to each one of them: $|+_{\theta_i}\rangle$, where θ_i is chosen at random from the set $A = \{0, \pi/4, \pi/2, 3\pi/4, \pi, 5\pi/5, 3\pi/2, 7\pi/4\}$ and sends them to Bob, who entangles them according to the graph. The protocol then follows the partial order given by the flow: Alice calculates the corrected measurement angle α'_i for each qubit using previous measurement results according to the flow dependences. She sends to Bob measurement angle $\delta_i = \alpha'_i + \theta_i + r_i\pi$, using an extra random bit r_i. Bob measures according to δ_i, reports the result back to Alice who corrects it by XOR-ing with r_i. In the case of quantum output, the final layer is sent to Alice and is also corrected according to the flow dependences by applying the corresponding Pauli operators.

Since the brickwork graph state satisfies the requirements of Theorem 7 we can adapt the Universal Blind Quantum Computing protocol by making Alice and Bob follow the order of Equation 13 and operate on input $|+\rangle\langle+| \otimes I_{w-1}/2^{w-1}$. A detailed description is given in Protocol 1.

▶ **Theorem 4.** *Protocol 1 is correct.*

Proof. Correctness comes from the fact that what Alice and Bob jointly compute is mathematically equivalent to performing the pattern of Equation 13 on input $|+\rangle\langle+| \otimes I_{w-1}/2^{w-1}$. The argument is the same as in the original universal blind quantum computing protocol [8]

Protocol 1 Blind BQ1P protocol

Alice's input:

- A vector of angles $\boldsymbol{a} = (a_{1,1}, \ldots, a_{w,d})$, where $a_{i,j}$ comes from the set $A = \{0, \pi/4, 2\pi/4, \ldots, 7\pi/4\}$, that when plugged in the measurement pattern $P_{\boldsymbol{a}}$ of Equation 13 applied on the brickwork state, implements the desired computation. This computation is applied on a fixed input state $|+\rangle\langle+| \otimes I_{w-1}/2^{w-1}$.

Alice's output:

- The top output qubit (qubit in position $(1, d)$).

The protocol

1. Alice picks a random angle $\theta_{1,1} \in A$, prepares one pure qubit in state $R_z(\theta_{1,1})|+\rangle$ and sends it to Bob who tags it as qubit $(1, 1)$.
2. Bob prepares the rest of input state (qubits $(2, 1), \ldots, (w, 1)$) in the maximally mixed state $I_{w-1}/2^{w-1}$.
3. Alice and Bob execute the rest of the computation in rounds. For $j = 1$ to $d - 1$ and for $i = 1$ to w
   a. **Alice's preparation**
      i. Alice picks a random angle $\theta_{i,j+1} \in A$.
      ii. Alice prepares one pure qubit in state $R_z(\theta_{i,j+1})|+\rangle$.
      iii. Alice sends it to Bob. Bob tags it as qubit $(i, j + 1)$.
   b. **Entanglement and measurement**
      i. Bob performs the entangling operator(s):

$$\prod_{\{k,l:(k,l)\sim(i,j),k\geq i,l\geq j\}} E_{(i,j),(k,l)}$$

      ii. Bob performs the rest of the computation using classical help from Alice:
         A. Alice computes the corrected measurement angle $a'_{i,j} = (-1)^{S^x_{i,j}} a_{i,j} + S^z_{i,j}\pi$.
         B. Alice chooses a random bit $r_{i,j}$ and computes $\delta_{i,j} = a'_{i,j} + \theta_{i,j} + r_{i,j}\pi$.
         C. Alice transmits $\delta_{i,j}$ to Bob.
         D. Bob performs operation $M^{\delta_{i,j}}_{i,j}$ which measures and traces over the qubit (i, j) and retrieves result $b_{i,j}$.
         E. Bob transmits $b_{i,j}$ to Alice.
         F. Alice updates the result to $s_{i,j} = b_{i,j} + r_{i,j} \mod 2$.
4. Bob sends to Alice the final layer of qubits, Alice performs the required corrections and outputs the result.

repeated here for completeness. Firstly, since entangling operators commute with R_z operators, preparing the pure qubits in a rotated state does not change the underlying graph state; only the phase of each qubit is locally changed, and it is as if Bob had performed the R_z rotation after the entanglement. Secondly, since a measurement in the $|+_a\rangle, |-_a\rangle$ basis on a state $|\phi\rangle$ is the same as a measurement in the $|+_{a+\theta}\rangle, |-_{a+\theta}\rangle$ basis on $R_z(\theta)|\phi\rangle$, and since $\delta = a' + \theta + \pi r$, if $r = 0$, Bob's measurement has the same effect as Alice's target measurement; if $r = 1$, all Alice needs to do is flip the outcome. ◀

Note that Protocol 1 can be trivially simplified by omitting all the measurements that are applied on maximally mixed states (i.e. all measurements applied on qubits in rows 2 to w from the beginning of the computation until each one is entangled with a non-maximally mixed qubit). However this does not give any substantial improvement in the complexity of the protocol.

▶ **Theorem 5.** *Protocol 1 is blind.*

(Proof Sketch). A detailed proof is provided in Appendix A. Intuitively, rotation by angle $\theta_{i,j}$ serves the purpose of hiding the actual measurement angle, while rotation by $r_{i,j}\pi$ hides the result of measuring the quantum state. This proof is consistent with definition of blindness based on the relation of Bob's system to Alice's system which takes into account prior knowledge of the secret and is a good indicator that blindness can be composable [14]. ◀

Regarding the complexity of the protocol, Alice needs to pick a polynomially large number of random bits and perform polynomially large number of modulo additions that is to say Alice classical computation is restricted to the class BPP. However Alice's quantum requirement is only to prepare single qubits, she has access to no quantum memory or quantum operation. Therefore assuming $BQ1P \not\subset BPP$ suggests Alice's quantum power is more restricted than $BQ1P$ and hence DQC1-MBQC. On the other hand, Bob performs a pattern of the form given in Equation 13, with the difference that instead of preparing the pure qubits himself, he receives the pure qubits through the quantum channel that connects him with Alice. Also, the qubits are not prepared in state $|+\rangle$, but in some state on the (X, Y) plane, but this doesn't alter the reasoning in the complexity proofs. Thus, Bob has computational power that is within the DQC1-MBQC complexity class according to the Corollary 13 and Theorem 7.

3 Verification

VBQC protocol is based on the ability to hide a trap qubit inside the graph state while not affecting the correct execution of the pattern. Both the trap qubit and the qubits which participate in the actual computation are prepared in the (X, Y) plane of the Bloch sphere. To keep them disentangled, some qubits (called dummy) prepared in the computational basis $\{|0\rangle, |1\rangle\}$, are injected between them. Being able to choose between the two states is essential for blindness (Theorem 4 in [18]). In particular, if a dummy qubit is in state $|0\rangle$, applying the entangling operator cZ between this qubit and a qubit prepared on the (X, Y) plane has no effect. If a dummy qubit is in state $|1\rangle$ then applying cZ will introduce a Pauli Z rotation on the qubit prepared on the (X, Y) plane. This effect can be cancelled by Alice in advance, by introducing a Pauli Z rotation on all the neighbours of $|1\rangle$'s when preparing the initial state.

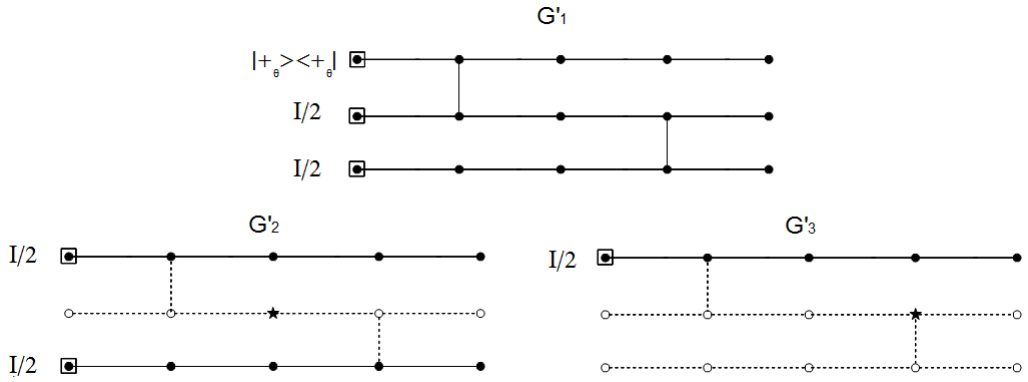

Figure 3 Let G' be the graph which consists of s isolated brickwork graphs (each denoted as G'_i), each of the same dimensions required for the desired computation. An example construction with $s = 3$ and one trap per graph together with a small brickwork state for computation is given above. Black vertices correspond to auxiliary qubits prepared on the $(X - Y)$ plane or mixed state when they are inputs (inside square), star vertices correspond to trap qubits and white vertices to auxiliary qubits prepared in the computational basis. Edges represent entangling operators, dashed where entangling has no effect (except of local rotations).

In the simplest version of VUBQC, a single trap, prepared in state $|+_{\theta_t}\rangle$, where θ is chosen at random from the angles set A (defined above) and placed at position t, chosen at random between all the vertices of the open graph state (G, I, O). During the execution of the pattern, if $t \notin O$, Bob is asked to measure qubit t with angle $\theta_t + r\pi$ and return the classical result b_t to Alice. If $b_t = r_t$ Alice sets an indicator bit to state acc (which means that this computation is accepted), otherwise she sets it to rej (computation is rejected). If $t \in O$, Alice herself measures the trap qubit and sets the indicator qubit accordingly. This version of the protocol is proven to be correct and ϵ-verifiable, with $\epsilon = (m - 1)/m$, where m is the size of the computation.

A generalisation of this technique which allows for arbitrary selection of parameter ϵ is also presented in [18]. By allowing for a polynomial number of traps to be injected in the graph state and adapting the computation inside a fault tolerant scheme with parameter d one can have ϵ inversely exponential to d. The question is whether this amplification method can also be used to design a verification protocol for DQC1-MBQC with arbitrary small ϵ. Unfortunately the underlying graph state used by this protocol does not have flow and not all qubits are prepared in the (X, Y) plane, so that one can not apply Theorem 7 to get a compatible rewriting of the pattern. Moreover, having the requirement that we should be able to place every trap qubit (which is a pure qubit) at any position in the graph, means that there exist patterns that will never be possible to be rewritten to satisfy the purity requirement. This leads us to seek a different approach for probability amplification for verification in the DQC1-MBQC model.

Instead of placing a polynomial number of isolated traps within the same graph, which is also used to perform the actual computation, we utilise s isolated brickwork subgraphs, one used for the computation and the rest being trap subgraphs (see Figure 3). Therefore at the beginning of the protocol, Alice chooses random parameter t_g, which denotes which graph will be the computational subgraph, and for each of the remaining trap subgraphs i, she chooses a random position t_i to hide one isolated trap. The rest of each trap subgraph will be a trivial computation (all measurement angles set to 0) on a totally mixed state,

and a selected set of dummy qubits are placed to isolate this computation from the trap. Computation subgraph and trap subgraphs are of the same size, and by taking advantage of the blindness of the protocol, Bob cannot distinguish between them. Therefore, to be able to cheat, he needs to deviate from the correct operation only during the execution on the computational subgraph and never deviate while operating on any of the traps. This gives the desirable ϵ parameter that will be proved later. The full description of protocol is given in Protocol 2. Each isolated pattern k is executed separately and according to the DCQ1-MBQC rewriting on the brickwork state given in Equation 13 in the blind setting. Pre-rotations on the neighbours of dummy qubits guarantee that the computation is not affected by the choice of dummies as described before. To prove the complexity of the protocol we need to notice that although the graph used satisfies the conditions of Theorem 7, the existence of the dummy qubits prepared in the computational basis creates the need of a new proof.

▶ **Theorem 6.** *The computational power of Bob in Protocol 2 is within DQC1-MBQC.*

Proof. Note that the s patterns are executed in series and Bob does not keep any qubits between executions. The inputs to these patterns are almost maximally mixed, in accordance with the purity requirement and this 'mixedness' propagates through both computational and trap subgraphs. For the computational subgraph (which is not entangled with the rest) the reasoning of the proof of Theorem 7 applies, since this subgraph satisfies the sufficient conditions and no dummy qubits are used. In the case of a trap subgraph k consider first those operations that apply on the isolated trap and dummy subgraph only. Then for each step $(i,j)_k$ of the main iteration of the protocol (where $(i,j)_k$ is a trap or a dummy) a new pure qubit is sent to Bob, which increases the purity parameter by 1. Entangling will not have any effect on the purity parameter. While the measurement does not increase the purity of the qubit since it was already pure (dummy or trap remain always pure through the computation), and tracing out the resulting qubit will decrease the purity by 1. Thus, the whole step will not change the purity. On the other hand, for the remaining operations the reasoning of the proof of Theorem 7 goes through, since this subgraph satisfies the sufficient conditions. Also operations that apply on both subgraphs are all unitaries therefore they do not affect purity. ◀

Using the definition of verifiability given in Definition 10 we prove the main theorem for the existence of a correct and verifiable DQC1-MBQC protocol (Theorem 11). The full proof is given in Appendix B, while here we describe the main steps.

Proof of Theorem 11 (Sketch). Correctness of Protocol 2 comes from the fact that the computational subgraph is disentangled from the rest of the computation and if Bob performs the predefined operations, from the correctness of the blind protocol Alice will receive the correct output. Also, in this case, (and since the traps are corrected to cancel the effect of their entanglement with their neighbouring dummies) the measurement of the traps will give the expected result and Alice will accept the computation.

The proof of verifiability follows the same general methodology of the proof of the original VUBQC protocol [18], except the last part which contains the counting arguments. For the rest we use single indexing for the qubits, where subgraph G'_i consists of m qubits indexed $(i-1)+1$ to im. Therefore the total number of qubits in the protocol is sm. Parameter n represents the size of the input of each subgraph (parameter w in the protocol).

Based on Definition 10 we need to bound the probability of the (purified) output collapsing onto the wrong subspace and accepting that result. To explicitly write the final state $B_j(\nu)$

Protocol 2 Verifiable DQC1-MBQC protocol with $s-1$ trap computations

Alice's input:

- An angle vector $\boldsymbol{a} = (a_{1,1}, \ldots, a_{w,d-1})$, where $a_{i,j}$ comes from the set $A = \{0, \pi/4, 2\pi/4, \ldots, 7\pi/4\}$, that, when plugged in the measurement pattern $P_{\boldsymbol{a}}$ of Equation 13 on the brickwork open graph state G of dimension (w,d) and flow $(f, \preceq)$, it implements the desired computation on fixed input $|+\rangle\langle+| \otimes I_{w-1}/2^{w-1}$.

Alice's output:

- The top output qubit of G (qubit in position $(1,d)$ in G) together with a 1-bit, named *acc*, that indicates if the result is accepted or not.

The protocol

- **Preparation steps.** Alice picks t_g at random from $\{1, \ldots, s\}$. Let G' be the graph which consists of s isolated brickwork graphs, each of the dimension the same as G. Then the t_g-th isolated graph (named G'_{t_g}) will be the computational subgraph for this run of the protocol.
- Alice maps the measurement angles of the computational subgraph G'_{t_g} to angles of graph G: $\boldsymbol{a}'_{G_{t_g} \setminus O_{t_g}} = \boldsymbol{a}$ and appropriately set the dependency sets S^x and S^z for all the vertices of G'_{t_g} (according to the standard flow), while for the rest of the vertices (graph $G' \setminus G'_{t_g}$) the sets S^x and S^z are empty.
- For $k = 1$ to s except t_g:
  1. Alice chooses one random vertex $\boldsymbol{t}_k = (t_x, t_y)_k$ among all vertices of G'_k for placing the trap.
  2. By G'_k's geometry, vertex (t_x, t_y) may be connected by a vertical edge to vertex (t'_x, t_y), where t'_x represents either $t_x + 1$ or $t_x - 1$. We add in D (set of dummies) all vertices of rows t_x, t'_x (if it exists) of G'_k, except the trap itself.
  3. All elements of $\boldsymbol{a}'_{G_k}$ are mapped to 0.
- Alice chooses random variables $\boldsymbol{\theta}_{G' \setminus D}$, each uniformly at random from A.
- Alice chooses random variables $\boldsymbol{r}_{G'}$ and $\boldsymbol{d}_D$, each uniformly at random from $\{0,1\}$.
- For $k = 1$ to s:
  1. **Initial step.** If $k = t_g$ then: Let $(1,1)_k$ be the position of the top input qubit in G'_k. Alice prepares the following states and sends them to Bob:

$$\begin{array}{rr} \{(1,1)_k\} & \left|+_{\theta_{(1,1)_k}}\right\rangle \\ \forall (i,1)_k \notin \{(1,1)_k\} & I/2 \end{array}$$

  Otherwise: Alice prepares the following states and sends them to Bob:

$$\begin{array}{rr} \forall (i,1)_k \in D & \left|d_{(i,1)_k}\right\rangle \\ (i,1)_k = \boldsymbol{t}_k & \prod_{\{m,l:(m,l)_k \sim (i,1)_k, (m,l)_k \in D\}} Z^{d_{(m,l)_k}} \left|+_{\theta_{(i,1)_k}}\right\rangle \\ \forall (i,1)_k \notin \{D, \boldsymbol{t}_k\} & I/2 \end{array}$$

Protocol 2 (cont'd)

2. **Main Iteration.** For $j = 1$ to $d-1$, for $i = 1$ to w:
   a. **Alice's preparation**
      i. Alice prepares one pure qubit in one of the following states, depending on $(i, j+1)_k$:
         $(i, j+1)_k \in D$ $\quad |d_{(i,j+1)_k}\rangle$
         $(i, j+1)_k \notin D$ $\quad \prod_{\{m,l:(m,l)_k \sim (i,j+1)_k, (m,l)_k \in D\}} Z^{d_{(m,l)_k}} |+_{\theta_{(i,j+1)_k}}\rangle$
      ii. Alice sends it to Bob. Bob labels it as qubit $(i, j+1)_k$.
   b. **Entanglement and measurement**
      i. Bob performs the entangling operator(s):

$$\prod_{\{m,l:(m,l)_k \sim (i,j)_k, m \geq i, l \geq j\}} E_{(i,j)_k,(m,l)_k}$$

      ii. Bob performs the rest of the computation using classical help from Alice:
         A. Alice computes the corrected measurement angle $a''_{(i,j)_k} = (-1)^{S^x_{(i,j)_k}} a'_{(i,j)_k} + S^z_{(i,j)_k}\pi$.
         B. Alice computes actual measurement angle $\delta_{(i,j)_k} = a''_{(i,j)_k} + \theta_{(i,j)_k} + r_{(i,j)_k}\pi$.
         C. Alice transmits $\delta_{(i,j)_k}$ to Bob.
         D. Bob performs operation $M^{\delta_{(i,j)_k}}_{(i,j)_k}$ which measures and traces over the qubit $(i, j)_k$ and retrieves result $b_{(i,j)_k}$.
         E. Bob transmits $b_{(i,j)_k}$ to Alice.
         F. Alice updates the result to $s_{(i,j)_k} = b_{(i,j)_k} + r_{(i,j)_k} \mod 2$.
3. Bob sends the final layer to Alice and Alice applies the final corrections if needed (only in round t_g).
4. If the trap qubit is within the qubits received, Alice measures it with angle $\delta_{\boldsymbol{t}_k} = \theta_{\boldsymbol{t}_k} + r_{\boldsymbol{t}_k}\pi$ to obtain $b_{\boldsymbol{t}_k}$. Also, Alice discards all qubits received by Bob in this round except qubit $(1, d)_{t_g}$.

Alice outputs qubit in position $(1, d)_{t_g}$ and sets bit acc to 1 if $b_{\boldsymbol{t}_k} = r_{\boldsymbol{t}_k}$ for all k.

we need to define the following notations. Alice's chosen random parameters are denoted collectively by ν, a subset of those are related to the traps: ν_T including t_g, t_k's and θ_{t_k}'s for $k \in \{1,\ldots,s\} \setminus t_g$. Also $\nu_C = \{\nu \setminus \nu_T\}$. The projection onto the correct state for each trap t_k is denoted by $\left|\eta_{t_k}^{\nu_T}\right\rangle$, where $\left|\eta_{t_k}^{\nu_T}\right\rangle = \left|+_{\theta_{t_k}}\right\rangle$ when $t_k \in O_k$ and $\left|\eta_{t_k}^{\nu_T}\right\rangle = |r_{t_k}\rangle$ otherwise (since the trap has been already measured). $C_{\boldsymbol{r}}$ denotes the Pauli operators that map the output state of the computational subgraph to the correct one. $c_{\boldsymbol{r}}$ is used to compactly deal with the fact that in the protocol each measured qubit i is decrypted by XOR-ing them with r_i, except for the trap qubits which remain uncorrected: $\forall k : (c_{\boldsymbol{r}})_{t_k} = 0$. $\rho_{M_k^\nu}$ denotes the density matrix representing the total quantum state received by Bob from Alice for each round k of the protocol. A special case is the t_kth round where $\rho_{M_k^\nu}$ represents the total state received by Bob together with its purification (not known to Bob). The classical information received by Bob at each elementary step i (measurement angles) are represented by $|\delta_i\rangle$'s.

We allow Bob to have an arbitrary deviation strategy j, at each elementary step i which is represented as CPTP map $\mathcal{E}_i^j$, followed by a Pauli Z measurement of qubit i (since Bob has to produce a classical bit at each step and return it to Alice), which is represented by taking the sum over projectors on the computational basis $|b_i\rangle$, for $b_i \in \{0,1\}$. All measurement operators can be commuted to the end of the computation and all CPTP maps can be gathered to a single map $\mathcal{E}^j$ after Bob has received everything from Alice, so that the failure probability can be written as:

$$p_{incorrect} = \sum_{\boldsymbol{b'},\nu} p(\nu)\mathrm{Tr}(P_\perp \bigotimes_{k=1}^{s} \left|\eta_{t_k}^{\nu_T}\right\rangle\left\langle\eta_{t_k}^{\nu_T}\right|$$
$$C^{\boldsymbol{b'},\nu_C}|\boldsymbol{b'}+\boldsymbol{c^r}\rangle\langle\boldsymbol{b'}+\boldsymbol{c^r}|C^{\boldsymbol{b'},\nu_C\dagger}) \mathcal{E}^j \left(\bigotimes_{k=1}^{s}\bigotimes_{i=1}^{m-n} \left|\delta_{(k-1)m+i}^{\boldsymbol{b'},\nu}\right\rangle\left\langle\delta_{(k-1)m+i}^{\boldsymbol{b'},\nu}\right| \otimes \rho_{M_k^\nu}\right) |\boldsymbol{b'}\rangle\langle\boldsymbol{b'}+\boldsymbol{c^r}|C^{\boldsymbol{b'},\nu_C\dagger})$$

Our strategy will be to rewrite this probability by introducing the correct execution of the protocol before the attack, on each subgraph k: $\mathcal{P}_k = \bigotimes_{i=1}^{m-n}(H_{(k-1)m+i}Z_{(k-1)m+i}(\delta_{(k-1)m+i}))E_{G'_k}$ and at the same time decomposing the attack to the Pauli basis, using general Paulis $\sigma_{i,k}$ applying on qubits $(k-1)m+1 \le \gamma \le km$ for each k.

$$p_{incorrect} = \sum_{\boldsymbol{b'},\nu,v,i,j} \alpha_{vi}\alpha_{vj}^* p(\nu)\mathrm{Tr}(P_\perp \bigotimes_{k=1}^{s} \left|\eta_{t_k}^{\nu_T}\right\rangle\left\langle\eta_{t_k}^{\nu_T}\right| C^{\boldsymbol{b'},\nu_C}|\boldsymbol{b'}+\boldsymbol{c^r}\rangle\langle\boldsymbol{b'}|$$
$$\bigotimes_{k=1}^{s}(\sigma_{i,k} \left(\mathcal{P}_k \bigotimes_{i=1}^{m-n} \left|\delta_{(k-1)m+i}^{\boldsymbol{b'},\nu}\right\rangle\left\langle\delta_{(k-1)m+i}^{\boldsymbol{b'},\nu}\right| \otimes \rho_{M_k^\nu}\mathcal{P}_k^\dagger\right) \sigma_{j,k})|\boldsymbol{b'}\rangle\langle\boldsymbol{b'}+\boldsymbol{c^r}|C^{\boldsymbol{b'},\nu_C\dagger}$$

This way we can characterise which Pauli attacks give non-zero failure probability when the final state is projected on the correct one. For convenience we introduce the following sets for an arbitrary Pauli $\sigma_{i,k}$:

$$A_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = I \text{ and } (k-1)m+1 \le \gamma \le km\}$$
$$B_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = X \text{ and } (k-1)m+1 \le \gamma \le km\}$$
$$C_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = Y \text{ and } (k-1)m+1 \le \gamma \le km\}$$
$$D_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = Z \text{ and } (k-1)m+1 \le \gamma \le km\}$$

We use the superscript O to denote subsets subject to the constraint $km \ge \gamma \ge km-n+1$. For an arbitrary t_g, the only attacks that give the corresponding term of the sum not equal to zero: are those that (i) produce an incorrect measurement result for qubits $(t_g-1)m+1 \le \gamma \le$

$t_g m - n$ or (ii) operate non-trivially on qubits $t_g m - n < \gamma \leq t_g m$. We denote this condition by $i \in E_{i,t_g}$ and $j \in E_{j,t_g}$: $|B_{i,t_g}| + |C_{i,t_g}| + |D^O_{i,t_g}| \geq 1$ and $|B_{j,t_g}| + |C_{j,t_g}| + |D^O_{j,t_g}| \geq 1$.

The next step will be to characterise which attacks of these subsets remain undetected by the trap mechanism and try to find an upper bound on their contribution to the failure probability. By applying blindness and observing that only the terms where $\sigma_{i,k} = \sigma_{j,k}$ contribute we obtain the following upper bound (details in Appendix B):

$$p_{incorrect} \leq \sum_{t_g} \sum_{v,i \in E_{i,t_g}} |\alpha_{vi}|^2 p(t_g) \prod_{k=\{1,\ldots,s\}\setminus t_g} \left(\sum_{\substack{km-n<t_k\leq km, \\ \theta_{t_k}}} p(t_k,\theta_{t_k}) (\left\langle +_{\theta_{t_k}} \right| \sigma_{i|t_k} \left| +_{\theta_{t_k}} \right\rangle \right)^2 + \sum_{\substack{(k-1)m<t_k\leq km-n, \\ r_{t_k}}} p(t_k, r_{t_k}) (\langle r_{t_k}|\sigma_{i|t_k}|r_{t_k}\rangle)^2)$$

The rest is based on a counting argument using $\forall k$, $|A_{i,k}| + |B_{i,k}| + |C_{i,k}| + |D_{i,k}| = m$.

$$p_{incorrect} \leq \sum_{t_g} \sum_{v,i \in E_{i,t_g}} |\alpha_{vi}|^2 \frac{1}{s} \prod_{k=\{1,\ldots,s\}\setminus t_g} \frac{1}{2m} (2|A_{i,k}| + |B^O_{i,k}| + |C^O_{i,k}| + 2|D_{i,k} \setminus D^O_{i,k}|)$$

$$\leq \sum_{t_g} \sum_{v,i \in E_{i,t_g}} |\alpha_{vi}|^2 \frac{1}{s} \prod_{k=\{1,\ldots,s\}\setminus t_g} \frac{1}{2m} (2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|)$$

We denote the product term $\prod_{k=\{1,2,3,\ldots,s\}\setminus z} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|)$ as $P_{i,z}$. We also denote each set $\{E^*_{i,1} \cap E^*_{i,2} \cap \ldots \cap E^*_{i,s}\}$, where each term $E^*_{i,w}$ is either $E_{i,w}$ or its complement, $E^C_{i,w}$, depending on whether the w-th value of a binary vector $\boldsymbol{y}$ (size s) is 1 or 0 respectively, as $W_{i,\boldsymbol{y}}$. Let the function $\#\boldsymbol{y}$ give the number of positions i such that y_i=1.

$$= \frac{1}{s} (\sum_{k=1}^{s} \sum_{\{\boldsymbol{y}:\#\boldsymbol{y}=k\}} \sum_{i \in W_{i,\boldsymbol{y}},v} (|\alpha_{vi}|^2 \sum_{\{z:y_z=1\}} P_{i,z}))$$

The condition $i \in W_{i,\boldsymbol{y}}$ means that the following conditions hold together: $\{|B_{i,w}| + |C_{i,w}| + |D^O_{i,w}| \geq 1 : y_w = 1\}$,$\{|B_{i,w}| + |C_{i,w}| + |D^O_{i,w}| = 0 : y_w = 0\}$.

$$\leq \frac{1}{s} (\sum_{k=1}^{s} \sum_{\{\boldsymbol{y}:\#\boldsymbol{y}=k\}} \sum_{i \in W_{i,\boldsymbol{y}},v} |\alpha_{vi}|^2 k \left(\frac{2m-1}{2m} \right)^{k-1}) = \frac{1}{s} (\sum_{k=1}^{s} c_k k \left(\frac{2m-1}{2m} \right)^{k-1})$$

where $c_k = \sum_{\{\boldsymbol{y}:\#\boldsymbol{y}=k\}} \sum_{i \in W_{i,\boldsymbol{y}},v} |\alpha_{vi}|^2$.

An upper bound on the above expression is:

$$p_{incorrect} < \frac{2m}{s} \tag{14}$$

◀

4 Conclusion

In this paper we present the first study of the delegation of quantum computing in a restricted model of computing and show that the general framework of the verification via blindness could be adapted to the setting of one-pure qubit model. In order to improve the obtained bound on the security parameter two open questions has to be addressed. The first one aims

to expand the class of resource states for DQC1 model so that several techniques from the MBQC domain could be applicable here. The second question will complement the first by searching for fault-tolerant schemes based on any new resource state for DQC1 model. More concretely we propose the study of following questions:

- A sufficient condition for compatibility with DQC1 based on the step-wise determinism criteria is presented in Theorem 7. Is this approach extendable to weaker notions of determinism such as information preserving maps as defined in [24]? Which is a necessary condition for a family of MBQC resource states to be universal for the DQC1 computation?
- Theorem 11 presents a scheme for verification where by adjusting the number of rounds one could obtain an ϵ-verifiable delegated DQC1-MBQC computing with ϵ being polynomially small on computation size. How can we efficiently amplify this bound to any desired exponentially small one? Is there a way to adapt the proposed probability amplification method of [18] based on a quantum error correcting code, into the DQC1-MBQC model?

Acknowledgements. We would like to thank Joe Fitzsimons, Vedran Dunjko and Alistair Stewart for useful discussions. AD was supported in part by EPSRC (Grant No. EP/H03031X/1), U.S. EOARD (Grant No. 093020), and the EU Integrated Project SIQS. TK was supported by Mary and Armeane Choksi Postgraduate Scholarship and School of Informatics Graduate School.

References

1 S. Aaronson and A. Arkhipov. The computational complexity of linear optics. In *STOC*, 2011.
2 D. Aharonov, M. Ben-Or, and E. Eban. Interactive proofs for quantum computations. In *Proceedings of Innovations in Computer Science 2010*, page 453, 2010.
3 H. Barnum, C. Crépeau, D. Gottesman, A. Smith, and A. Tapp. Authentication of quantum messages. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2002)*, page 449, 2002.
4 S. Barz, E. Kashefi, A. Broadbent, J. F. Fitzsimons, A. Zeilinger, and P. Walther. Demonstration of blind quantum computing. *Science*, 335(6066):303–308, 2012.
5 Stefanie Barz, Joseph F Fitzsimons, Elham Kashefi, and Philip Walther. Experimental verification of quantum computation. *Nature Physics*, 2013.
6 Sergey B. Bravyi and Alexei Yu. Kitaev. Fermionic quantum computation. *Annals of Physics*, 298:210, 2002.
7 M. Bremner, R. Jozsa, and D. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proc. Roy. Soc. A*, 467:459, 2011.
8 A. Broadbent, J. Fitzsimons, and E. Kashefi. Universal blind quantum computing. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009)*, page 517, 2009.
9 D. Browne, E. Kashefi, M. Mhalla, and S. Perdrix. Generalized flow and determinism in measurement-based quantum computation. *New Journal of Physics*, 9:250, 2007.
10 V. Danos and E. Kashefi. Determinism in the one-way model. *Physical Review A*, 74:052310, 2006.
11 V. Danos, E. Kashefi, and P. Panangaden. The measurement calculus. *Journal of ACM*, 54:8, 2007.
12 A. Datta and A. Shaji. Quantum discord and quantum computing – an appraisal. *International Journal of Quantum Information*, 9:1787, 2011.

13 D. DiVincenzo. The physical implementation of quantum computation. *Fortschr. Phys.*, 48:771, 2000.

14 V. Dunjko, J. Fitzsimons, C. Portmann R., and Renner. Composable security of delegated quantum computation. *arXiv preprint arXiv:1301.3662*, 2013.

15 V. Dunjko, E. Kashefi, and A. Leverrier. Universal blind quantum computing with coherent states. *arXiv preprint arXiv:1108.5571*, 2011.

16 Vedran Dunjko. Ideal quantum protocols in the non-ideal physical world. *PhD Thesis, Heriot-Watt University*, 2012.

17 B. Reichardt F., Unger, and U. Vazirani. Classical command of quantum systems. *Nature*, 496, 2013.

18 Joseph F Fitzsimons and Elham Kashefi. Unconditionally verifiable blind computation. *arXiv preprint arXiv:1203.5217*, 2012.

19 V. Giovannetti, L. Maccone, T. Morimae, and T. Rudolph. Efficient universal blind computation. *arXiv preprint arXiv:1306.2724*, 2013.

20 M. Hein, J. Eisert, and H. J. Briegel. Multi-party entanglement in graph states. *Physical Review A*, 69, 2004. quant-ph/0307130.

21 Stephen P. Jordan. Permutational quantum computing. *Quantum Info. Comput.*, 10(5):470–497, May 2010.

22 E. Knill and R. Laflamme. Power of one bit of quantum information. *Phys. Rev. Lett.*, 81:5672, 1998.

23 A. Mantri, C. Perez-Delgado, and J. Fitzsimons. Optimal blind quantum computation. *arXiv preprint arXiv:1306.3677*, 2013.

24 M. Mhalla, M. Murao, S. Perdrix, M. Someya, and P. Turner. Which graph states are useful for quantum information processing? In *TQC Theory of Quantum Computation, Communication and Cryptography 2011*, 2010.

25 T. Morimae, V. Dunjko, and E. Kashefi. Ground state blind quantum computation on aklt state. *arXiv preprint arXiv:1009.3486*, 2011.

26 Tomoyuki Morimae. Continuous-variable blind quantum computation. *Phys. Rev. Lett.*, 109:230502, Dec 2012.

27 Tomoyuki Morimae and Keisuke Fujii. Blind topological measurement-based quantum computation. *Nature Communications*, 3:1036, 2012.

28 Tomoyuki Morimae and Keisuke Fujii. Blind quantum computation protocol in which alice only makes measurements. *Phys. Rev. A*, 87:050301, May 2013.

29 Tomoyuki Morimae, Keisuke Fujii, and Joseph F Fitzsimons. On the hardness of classically simulating the one clean qubit model. *arXiv preprint arXiv:1312.2496*, 2013.

30 Anna Pappa, André Chailloux, Stephanie Wehner, Eleni Diamanti, and Iordanis Kerenidis. Multipartite entanglement verification resistant against dishonest parties. *Physical Review Letters*, 108(26), 2012.

31 R. Raussendorf and H. J. Briegel. A one-way quantum computer. *Physical Review Letters*, 86:5188–5191, 2001.

32 D. Shepherd. Computing with unitaries and one pure qubit. *arXiv:quant-ph/0608132*, 2006.

33 P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26:1484–1509, 1997. First published in 1995.

34 T. Sueki, T. Koshiba, and T. Morimae. Ancilla-driven universal blind quantum computation. *Physical Review A*, 87, 2013.

A Proof of Theorem 5

Proof. In this proof of blindness for Protocol 1 we use techniques developed in [16]. The basic difference from the proof of [16] arises from the different order in which Bob receives the states from Alice. Nevertheless, after commuting all CPTP maps into a single operator at the end, the methodology for proving blindness is the same as in the original proof. We give the full proof here for the sake of clarity.

To prove blindness we do not separate Alice's system into a classical and a quantum part but we consider the whole of Alice's system as quantum. This is a reasonable assumption since a classical system can be viewed as a special case of a quantum system. Therefore, by proving blindness for the more general case we also prove blindness for the special case.

For the sake of clarity we use single indexing for all the qubits of the resource state. The total number of qubits is denoted by m and the number of qubits in a single column of the brickwork state is denoted by n.

Our goal will be to explicitly write the state $\sigma_B = \mathrm{Tr}_A(\sigma_{AB})$ that Bob holds at the end of the execution of the protocol. To achieve this we express Bob's behaviour at each step i of the protocol as a collection of completely-positive trace-preserving (CPTP) maps $\mathcal{E}_i^{b_i}$, each for every possible classical response b_i from Bob to Alice.

At step 1 of the main loop of the protocol Bob has already been given the top input qubit at position 1 (position $(1,1)$ in the protocol notation) and the qubit at position $f(1) = 1+n$ (position $(1,2)$ in the protocol notation) together with the angle for measuring qubit 1 (angle can be represented as a quantum state composed of 3 qubits). State $\mathrm{Tr}_A(\rho_{AB})$ represents Bob's state before the protocol begins and can, in general, be dependent on Alice's secret measurement angles. The state of Bob averaged over all possible choices of Alice and possible classical responses from Bob, after step 1 is:

$$\sum_{b_1,r_1,\theta_1,\theta_{1+n}} \mathcal{E}_1^{b_1}\left(\left|\delta_1^{\theta_1,r_1}\right\rangle\!\left\langle\delta_1^{\theta_1,r_1}\right| \otimes \left|+_{\theta_{1+n}}\right\rangle\!\left\langle+_{\theta_{1+n}}\right| \otimes \left|+_{\theta_1}\right\rangle\!\left\langle+_{\theta_1}\right| \otimes \mathrm{Tr}_A(\rho_{AB})\right)$$

Note the all binary parameters in sums range over 0 and 1, ex. $\sum_{b_1}$ stands for $\sum_{b_1=0}^{1}$ and all angles range over the 8 possible values in A.

We can write the state of Bob after step 2 of the main iteration as:

$$\sum_{b_2,b_1,r_2,r_1,\theta_{2+n},\theta_{1+n},\theta_2,\theta_1} \mathcal{E}_2^{b_2}\left(\left|\delta_2^{\theta_2,r_2}\right\rangle\!\left\langle\delta_2^{\theta_2,r_2}\right| \otimes \left|+_{\theta_{2+n}}\right\rangle\!\left\langle+_{\theta_{2+n}}\right| \otimes \mathcal{E}_1^{b_1}\left(\left|\delta_1^{\theta_1,r_1}\right\rangle\!\left\langle\delta_1^{\theta_1,r_1}\right| \otimes \left|+_{\theta_{1+n}}\right\rangle\!\left\langle+_{\theta_{1+n}}\right| \otimes \left|+_{\theta_1}\right\rangle\!\left\langle+_{\theta_1}\right| \otimes \mathrm{Tr}_A(\rho_{AB})\right)\right)$$

Following this analysis, after the last step of the iteration Bob's state will be:

$$\sigma_B = \sum_{\substack{\boldsymbol{b}_{\le m-n},\\ \boldsymbol{r}_{\le m-n},\boldsymbol{\theta}_{\le m}}} \mathcal{E}_{m-n}^{b_{m-n}}\left(\left|\delta_{m-n}^{\boldsymbol{b}_{<m-n},\boldsymbol{r}_{\le m-n},\theta_{m-n}}\right\rangle\!\left\langle\delta_{m-n}^{\boldsymbol{b}_{<m-n},\boldsymbol{r}_{\le m-n},\theta_{m-n}}\right| \otimes \left|+_{\theta_m}\right\rangle\!\left\langle+_{\theta_m}\right|\right.$$
$$\otimes \ldots \otimes \mathcal{E}_2^{b_2}\left(\left|\delta_2^{\theta_2,r_2}\right\rangle\!\left\langle\delta_2^{\theta_2,r_2}\right| \otimes \left|+_{\theta_{2+n}}\right\rangle\!\left\langle+_{\theta_{2+n}}\right|\right.$$
$$\left.\left.\otimes \mathcal{E}_1^{b_1}\left(\left|\delta_1^{\theta_1,r_1}\right\rangle\!\left\langle\delta_1^{\theta_1,r_1}\right| \otimes \left|+_{\theta_{1+n}}\right\rangle\!\left\langle+_{\theta_{1+n}}\right| \otimes \left|+_{\theta_1}\right\rangle\!\left\langle+_{\theta_1}\right| \otimes \mathrm{Tr}_A(\rho_{AB})\right)\right) \ldots\right)$$

Notation $\boldsymbol{b}_{<m-n}$ stands for all the elements of $\boldsymbol{b}$ with index less than $m-n$.

Collecting all CPTP maps by commuting them with systems which they do not apply on

into a single operator $\mathcal{E}$ and rearranging the terms of the tensor product inside gives:

$$= \sum_{\substack{\boldsymbol{b}_{\leq m-n},\\ \boldsymbol{r}_{\leq m-n},\boldsymbol{\theta}_{\leq m}}} \mathcal{E}^{\boldsymbol{b}_{\leq m-n}}\Big(\bigotimes_{i=m-n}^{m} |+_{\theta_i}\rangle\langle+_{\theta_i}| \bigotimes_{i=n+1}^{m-n-1} (\left|\delta_i^{\boldsymbol{b}_{<i},\boldsymbol{r}_{\leq i},\theta_i}\right\rangle\left\langle\delta_i^{\boldsymbol{b}_{<i},\boldsymbol{r}_{\leq i},\theta_i}\right| \otimes |+_{\theta_i}\rangle\langle+_{\theta_i}|) \bigotimes_{i=2}^{n}(\left|\delta_i^{\theta_i,r_i}\right\rangle\left\langle\delta_i^{\theta_i,r_i}\right|) \otimes \left|\delta_1^{\theta_1,r_1}\right\rangle\left\langle\delta_1^{\theta_1,r_1}\right| \otimes |+_{\theta_1}\rangle\langle+_{\theta_1}| \otimes \mathrm{Tr}_A(\rho_{AB})\Big)$$

We introduce the controlled unitary:

$$U = \prod_{n+1\leq i\leq m-n-1, i=1} Z_i(-\delta_i)$$

and rewrite the state as:

$$\sum_{\substack{\boldsymbol{b}_{\leq m-n},\\ \boldsymbol{r}_{\leq m-n},\boldsymbol{\theta}_{\leq m}}} \mathcal{E}^{\boldsymbol{b}_{\leq m-n}}\Big(U^\dagger U \bigotimes_{i=m-n}^{m} |+_{\theta_i}\rangle\langle+_{\theta_i}| \bigotimes_{i=n+1}^{m-n-1} (\left|\delta_i^{\boldsymbol{b}_{<i},\boldsymbol{r}_{\leq i},\theta_i}\right\rangle\left\langle\delta_i^{\boldsymbol{b}_{<i},\boldsymbol{r}_{\leq i},\theta_i}\right| \otimes |+_{\theta_i}\rangle\langle+_{\theta_i}|) \bigotimes_{i=2}^{n}(\left|\delta_i^{\theta_i,r_i}\right\rangle\left\langle\delta_i^{\theta_i,r_i}\right|) \otimes \left|\delta_1^{\theta_1,r_1}\right\rangle\left\langle\delta_1^{\theta_1,r_1}\right| \otimes |+_{\theta_1}\rangle\langle+_{\theta_1}|U^\dagger U \otimes \mathrm{Tr}_A(\rho_{AB})\Big)$$

After applying the innermost unitary and absorbing the outermost into the CPTP-map we have:

$$\sum_{\substack{\boldsymbol{b}_{\leq m-n},\\ \boldsymbol{r}_{\leq m-n},\boldsymbol{\theta}_{\leq m}}} \mathcal{E}'^{\boldsymbol{b}_{\leq m-n}}\Big(\bigotimes_{i=m-n}^{m} |+_{\theta_i}\rangle\langle+_{\theta_i}| \bigotimes_{i=n+1}^{m-n-1} \left(\left|\delta_i^{\boldsymbol{b}_{<i},\boldsymbol{r}_{\leq i},\theta_i}\right\rangle\left\langle\delta_i^{\boldsymbol{b}_{<i},\boldsymbol{r}_{\leq i},\theta_i}\right| \otimes \left|+_{-a_i'^{\,\boldsymbol{b}_{<i},\boldsymbol{r}_{<i}}-r_i\pi}\right\rangle\left\langle+_{-a_i'^{\,\boldsymbol{b}_{<i},\boldsymbol{r}_{<i}}-r_i\pi}\right|\right) \bigotimes_{i=2}^{n} \left(\left|\delta_i^{\theta_i,r_i}\right\rangle\left\langle\delta_i^{\theta_i,r_i}\right|\right) \otimes \left|\delta_1^{\theta_1,r_1}\right\rangle\left\langle\delta_1^{\theta_1,r_1}\right| \otimes |+_{-a_1'-r_1\pi}\rangle\langle+_{-a_1'-r_1\pi}| \otimes \mathrm{Tr}_A(\rho_{AB})\Big)$$

It is essential for the proof that each term with index i in the tensor products depends only on parameters with index $\leq i$. This allows to break the summations over $\boldsymbol{r}_{\leq m-n}$ and $\boldsymbol{\theta}_{\leq m}$ and calculate them iteratively from left to right, given the following:

$$\sum_{\theta_i} |+_{\theta_i}\rangle\langle+_{\theta_i}| = \frac{I_1}{2}$$

where $I_n = \bigotimes_n I$. Also,

$$\begin{aligned}
&\sum_{r_i,\theta_i} \left|\delta_i^{\boldsymbol{r}_{\leq i},\theta_i}\right\rangle\left\langle\delta_i^{\boldsymbol{r}_{\leq i},\theta_i}\right| \otimes \left|+_{-a_i'^{\,\boldsymbol{r}_{<i}}-r_i\pi}\right\rangle\left\langle+_{-a_i'^{\,\boldsymbol{r}_{<i}}-r_i\pi}\right| \\
&= \sum_{r_i}\Big(\sum_{\theta_i}\left(\left|a_i'^{\,\boldsymbol{r}_{<i}}+\theta_i+r_i\pi\right\rangle\left\langle a_i'^{\,\boldsymbol{r}_{<i}}+\theta_i+r_i\pi\right|\right) \otimes \left|+_{-a_i'^{\,\boldsymbol{r}_{<i}}-r_i\pi}\right\rangle\left\langle+_{-a_i'^{\,\boldsymbol{r}_{<i}}-r_i\pi}\right|\Big) \\
&= \sum_{r_i} \frac{I_3}{2^3} \otimes \left|+_{-a_i'^{\,\boldsymbol{r}_{<i}}-r_i\pi}\right\rangle\left\langle+_{-a_i'^{\,\boldsymbol{r}_{<i}}-r_i\pi}\right| \\
&= \frac{I_4}{2^4}
\end{aligned}$$

and

$$\sum_{r_i,\theta_i} \left|\delta_i^{\theta_i,r_i}\right\rangle\left\langle\delta_i^{\theta_i,r_i}\right| = \frac{I_3}{2^3}$$

This procedure will produce the state:

$$\sigma_B = \mathcal{E}'\left(\frac{I_{4m-4n+1}}{2^{4m-4n+1}} \otimes \mathrm{Tr}_A(\rho_{AB})\right) = \mathcal{E}''(\mathrm{Tr}_A(\rho_{AB}))$$

where $\mathcal{E}''$ is some CPTP map. Therefore Definition 8 is satisfied. ◀

B Proof of Theorem 11

Proof. The same notation is used as in Section 3. The first step is to write the state of Alice's system at the end of the execution of the protocol for fixed Bob's behaviour j and choices of Alice ν. We have utilised the fact that all measurements can be moved to the end. Also, we have commuted all Bob's operations to the end (before the measurements) merging them to a single CPTP map. The state of Alice is:

$$B_j(\nu) = \sum_{\boldsymbol{b}} \otimes_{i=k}^{s} \left|+_{\theta_{t_k}+b_{t_k}\pi}\right\rangle\left\langle+_{\theta_{t_k}+b_{t_k}\pi}\right| C^{\boldsymbol{b},\nu_C}|\boldsymbol{b}+\boldsymbol{c}^{\boldsymbol{r}}\rangle\langle\boldsymbol{b}|$$
$$\mathcal{E}^j\left(\bigotimes_{k=1}^{s}\bigotimes_{i=1}^{m-n}\left|\delta^{\boldsymbol{b},\nu}_{(k-1)m+i}\right\rangle\left\langle\delta^{\boldsymbol{b},\nu}_{(k-1)m+i}\right| \otimes \rho_{M_k^\nu}\right)|\boldsymbol{b}\rangle\langle\boldsymbol{b}+\boldsymbol{c}^{\boldsymbol{r}}|C^{\boldsymbol{b},\nu_C\dagger}\otimes_{i=k}^{s}\left|+_{\theta_{t_k}+b_{t_k}\pi}\right\rangle\left\langle+_{\theta_{t_k}+b_{t_k}\pi}\right|$$

where $\left|+_{\theta_{t_k}+b_{t_k}\pi}\right\rangle\left\langle+_{\theta_{t_k}+b_{t_k}\pi}\right|$ are used to define Alice's measurement of the traps which are part of the output state of each round k (if they exist).

To bound the failure probability, observe that projectors orthogonal to $\left|\eta_{t_k}^{\nu_T}\right\rangle$'s vanish, thus we have (where $b' = \{b_i\}_{i\neq t_1\ldots t_s}$):

$$p_{incorrect} = \sum_{\boldsymbol{b'},\nu} p(\nu)\mathrm{Tr}(P_\perp \bigotimes_{k=1}^{s}\left|\eta_{t_k}^{\nu_T}\right\rangle\left\langle\eta_{t_k}^{\nu_T}\right|$$
$$C^{\boldsymbol{b'},\nu_C}|\boldsymbol{b'}+\boldsymbol{c}^{\boldsymbol{r}}\rangle\langle\boldsymbol{b'}+\boldsymbol{c}^{\boldsymbol{r}}|C^{\boldsymbol{b'},\nu_C\dagger})\mathcal{E}^j\left(\bigotimes_{k=1}^{s}\bigotimes_{i=1}^{m-n}\left|\delta^{\boldsymbol{b'},\nu}_{(k-1)m+i}\right\rangle\left\langle\delta^{\boldsymbol{b'},\nu}_{(k-1)m+i}\right| \otimes \rho_{M_k^\nu}\right)|\boldsymbol{b'}\rangle\langle\boldsymbol{b'}+\boldsymbol{c}^{\boldsymbol{r}}|C^{\boldsymbol{b'},\nu_C\dagger})$$

We introduce the following unitary, which characterises the correct operation on each subgraph k: $\mathcal{P}_k = \bigotimes_{i=1}^{m-n}(H_{(k-1)m+i}Z_{(k-1)m+i}(\delta_{(k-1)m+i}))E_{G'_k}$.

We can rewrite the failure probability, introducing $\mathcal{P}_k^\dagger\mathcal{P}_k$'s on both sides of the quantum state of the system before the attack, and absorbing the outermost unitaries into the updated CPTP map $\mathcal{E}'^j$:

$$p_{incorrect} = \sum_{\boldsymbol{b'},\nu} p(\nu)\mathrm{Tr}(P_\perp \bigotimes_{k=1}^{s}\left|\eta_{t_k}^{\nu_T}\right\rangle\left\langle\eta_{t_k}^{\nu_T}\right| C^{\boldsymbol{b'},\nu_C}$$
$$|\boldsymbol{b'}+\boldsymbol{c}^{\boldsymbol{r}}\rangle\langle\boldsymbol{b'}|\mathcal{E}'^j\left(\bigotimes_{k=1}^{s}(\mathcal{P}_k\bigotimes_{i=1}^{m-n}\left|\delta^{\boldsymbol{b'},\nu}_{(k-1)m+i}\right\rangle\left\langle\delta^{\boldsymbol{b'},\nu}_{(k-1)m+i}\right| \otimes \rho_{M_k^\nu}\mathcal{P}_k^\dagger)\right)|\boldsymbol{b'}\rangle\langle\boldsymbol{b'}+\boldsymbol{c}^{\boldsymbol{r}}|C^{\boldsymbol{b'},\nu_C\dagger})$$

We decompose $\mathcal{E}'^j$ using the following facts: There exist some matrices $\{\chi_v\}$ of dimension $s(4m-3n)\times s(4m-3n)$, with $\sum_v \chi_v\chi_v^\dagger = I$ such that for every density operator ρ:

$\mathcal{E}'^{j}(\rho) = \sum_v \chi_v \rho \chi_v^\dagger$. Also, each χ_v can be decomposed to the Pauli basis: $\chi_v = \sum_i \alpha_{vi}\sigma_i$, with $\sum_{v,i} \alpha_{vi}\alpha_{vi}^* = 1$. Setting $\sigma_{i,k}$ to be the part of σ_i that applies on the qubits $(k-1)m+1 \leq \gamma \leq km$.

$$p_{incorrect} = \sum_{\boldsymbol{b'},\nu,v,i,j} \alpha_{vi}\alpha_{vj}^* p(\nu) \mathrm{Tr}(P_\perp \bigotimes_{k=1}^{s} \left|\eta_{t_k}^{\nu_T}\right\rangle\!\left\langle\eta_{t_k}^{\nu_T}\right| C^{\boldsymbol{b'},\nu_C}$$
$$|\boldsymbol{b'}+\boldsymbol{c^r}\rangle\langle\boldsymbol{b'}| \bigotimes_{k=1}^{s} (\sigma_{i,k} \left(\mathcal{P}_k \bigotimes_{i=1}^{m-n} \left|\delta_{(k-1)m+i}^{\boldsymbol{b'},\nu}\right\rangle\!\left\langle\delta_{(k-1)m+i}^{\boldsymbol{b'},\nu}\right| \otimes \rho_{M_k^\nu} \mathcal{P}_k^\dagger \right) \sigma_{j,k}) |\boldsymbol{b'}\rangle\langle\boldsymbol{b'}+\boldsymbol{c^r}| C^{\boldsymbol{b'},\nu_C\dagger}$$

Without loss of generality we can assume that σ_i, σ_j do not change the δ's.

For an arbitrary t_g, the only attacks that give the corresponding term of the sum not equal to zero:

$$P_\perp(C^{\boldsymbol{b'},\nu_C}|\boldsymbol{b'}\rangle\langle\boldsymbol{b'}+\boldsymbol{c^r}|\sigma_{i,t_g}$$
$$(\mathcal{P}_{t_g} \bigotimes_{i=1}^{m-n} \left|\delta_{(t_g-1)m+i}^{\boldsymbol{b'},\nu}\right\rangle\!\left\langle\delta_{(t_g-1)m+i}^{\boldsymbol{b'},\nu}\right| \otimes \rho_{M_{t_g}^\nu} \mathcal{P}_{t_g}^\dagger)\sigma_{j,t_g} |\boldsymbol{b'}\rangle\langle\boldsymbol{b'}+\boldsymbol{c^r}| C^{\boldsymbol{b'},\nu_C\dagger}) \neq 0$$

are those that (i) produce an incorrect measurement result for qubits $(t_g-1)m+1 \leq \gamma \leq t_g m - n$ or (ii) operate non-trivially on qubits $t_g m - n < \gamma \leq t_g m$. We denote this condition by $i \in E_{i,t_g}$ and $j \in E_{j,t_g}$.

We can rewrite the probability by eliminating $P_\perp$ (observing that it applies to a positive operator) and $C^{\boldsymbol{b'},\nu_C}$ (by the cyclical property of the trace):

$$p_{incorrect} \leq \sum_{\nu,v,i\in E_{i,t_g},j\in E_{j,t_g}} \alpha_{vi}\alpha_{vj}^* p(\nu) \prod_{k=1}^{s} \mathrm{Tr}(\left|\eta_{t_k}^{\nu_T}\right\rangle\!\left\langle\eta_{t_k}^{\nu_T}\right|$$
$$|\boldsymbol{b'}\rangle\langle\boldsymbol{b'}+\boldsymbol{c^r}|\sigma_{i,k} \left(\mathcal{P}_k \bigotimes_{i=1}^{m-n} \left|\delta_{(k-1)m+i}^{\boldsymbol{b'},\nu}\right\rangle\!\left\langle\delta_{(k-1)m+i}^{\boldsymbol{b'},\nu}\right| \otimes \rho_{M_k^\nu} \mathcal{P}_k^\dagger \right) \sigma_{j,k})$$

We extract a trace over R from $\rho_{M_{t_g}^\nu}$. And extract the sums over $\nu_{C,k}$'s from the general sum, where $\nu_{C,k}$ is the subset of random parameters ν_C that are used for the computation of round r:

$$= \sum_{\nu_T,v,i\in E_{i,t_g},j\in E_{j,t_g}} \alpha_{vi}\alpha_{vj}^* p(\nu_T) \prod_{k=1}^{s} \mathrm{Tr}(\left|\eta_{t_k}^{\nu_T}\right\rangle\!\left\langle\eta_{t_k}^{\nu_T}\right|$$
$$|\boldsymbol{b'}\rangle\langle\boldsymbol{b'}+\boldsymbol{c^r}|\sigma_{i,k} \left(\mathcal{P}_k \sum_{\nu_{C,k}} (p(\nu_{C,k}) \bigotimes_{i=1}^{m-n} \left|\delta_{(k-1)m+i}^{\boldsymbol{b'},\nu}\right\rangle\!\left\langle\delta_{(k-1)m+i}^{\boldsymbol{b'},\nu}\right| \otimes \mathrm{Tr}_R(\rho_{M_k^\nu}))\mathcal{P}_k^\dagger \right) \sigma_{j,k})$$

To take advantage of the blindness property we use the following lemma where the proof is given later.

▶ **Lemma 7** (Blindness (excluding the traps))**.**

$$\forall k, \sum_{\nu_{C,k}} p(\nu_{C,k}) \bigotimes_{i=1}^{m-n} \left|\delta_{(k-1)m+i}^{\boldsymbol{b'},\nu}\right\rangle\!\left\langle\delta_{(k-1)m+i}^{\boldsymbol{b'},\nu}\right| \otimes Tr_R(\rho_{M_k^\nu})$$
$$= \frac{I_k^{t_k}}{Tr(I_k^{t_k})} \otimes \left|\delta_{t_k}^{\theta_{t_k},r_{t_k}}\right\rangle\!\left\langle\delta_{t_k}^{\theta_{t_k},r_{t_k}}\right| \otimes \left|+_{\theta_{t_k}}\right\rangle\!\left\langle+_{\theta_{t_k}}\right|$$

If $k \neq t_g$, $I_k^{t_k} = \bigotimes_{4m-3n-1} I$ *when* $km - n < t_k \leq km$ *and* $I_k^{t_k} = \bigotimes_{4m-3n-4} I$ *when* $(k-1)m < t_k \leq km - n$ *. And if* $k = t_g$, $I_k^{t_k} = \bigotimes_{4m-3n} I$.

Lemma 7 allows us to simplify the big sum above based on the position of the traps. We also sum over $\boldsymbol{b'}$ since there are no longer any dependencies on it in the sum, obtaining:

$$= \sum_{t_g, v, i\in E_{i,t_g}, j\in E_{j,t_g}} \alpha_{vi}\alpha^*_{vj} p(t_g) \prod_{k=1}^{s} \mathrm{Tr}(\sum_{\substack{km-n<t_k\le km, \\ \theta_{t_k}}} p(t_k,\theta_{t_k}) \Big|+_{\theta_{t_k}}\Big\rangle\Big\langle+_{\theta_{t_k}}\Big| \sigma_{i,k}(\frac{\mathcal{I}}{\mathrm{Tr}(\mathcal{I})} \otimes \Big|+_{\theta_{t_k}}\Big\rangle\Big\langle+_{\theta_{t_k}}\Big|)\sigma_{j,k} + \sum_{\substack{(k-1)m<t_k\le km-n, \\ r_{t_k}}} p(t_k,r_{t_k}) |r_{t_k}\rangle\langle r_{t_k}| \sigma_{i,k}(\frac{\mathcal{I}}{\mathrm{Tr}(\mathcal{I})} \otimes |r_{t_k}\rangle\langle r_{t_k}|)\sigma_{j,k})$$

where $\mathcal{I} = \bigotimes_{4m-3n-1} I$ when $k \neq t_g$. And $\mathcal{I} = \bigotimes_{4m-3n} I$ when $k = t_g$.

Note that $\sum_{\theta_{t_k}} \mathrm{Tr}(\Big|+_{\theta_{t_k}}\Big\rangle\Big\langle+_{\theta_{t_k}}\Big| \sigma_{i,k}(\frac{\mathcal{I}}{\mathrm{Tr}(\mathcal{I})} \otimes \Big|+_{\theta_{t_k}}\Big\rangle\Big\langle+_{\theta_{t_k}}\Big|)\sigma_{j,k})$ is zero if $\sigma_{i,k} \neq \sigma_{j,k}$. The same is true for $\sum_{r_{t_k}} \mathrm{Tr}(|r_{t_k}\rangle\langle r_{t_k}| \sigma_{i,k}(\frac{\mathcal{I}}{\mathrm{Tr}(\mathcal{I})} \otimes |r_{t_k}\rangle\langle r_{t_k}|)\sigma_{j,k})$. Therefore we can only keep those terms where $\sigma_{i,k} = \sigma_{j,k}$ and the failure probability becomes:

$$= \sum_{t_g} \sum_{v, i\in E_{i,t_g}} |\alpha_{vi}|^2 p(t_g) \prod_{k=\{1,\ldots,s\}\setminus t_g} (\sum_{\substack{km-n<t_k\le km, \\ \theta_{t_k}}} p(t_k,\theta_{t_k}) (\Big\langle+_{\theta_{t_k}}\Big| \sigma_{i|t_k} \Big|+_{\theta_{t_k}}\Big\rangle)^2 + \sum_{\substack{(k-1)m<t_k\le km-n, \\ r_{t_k}}} p(t_k,r_{t_k}) (\langle r_{t_k}|\sigma_{i|t_k}|r_{t_k}\rangle)^2)$$

The rest of the proof is based on a counting argument. For convenience we introduce the following sets for an arbitrary Pauli $\sigma_{i,k}$:

$$A_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = I \text{ and } (k-1)m+1 \le \gamma \le km\}$$
$$B_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = X \text{ and } (k-1)m+1 \le \gamma \le km\}$$
$$C_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = Y \text{ and } (k-1)m+1 \le \gamma \le km\}$$
$$D_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = Z \text{ and } (k-1)m+1 \le \gamma \le km\}$$

and use the superscript O to denote subsets subject to the constraint $km \ge \gamma \ge km-n+1$.

The failure probability is then:

$$= \sum_{t_g} \sum_{v, i\in E_{i,t_g}} |\alpha_{vi}|^2 \frac{1}{s} \prod_{k=\{1,\ldots,s\}\setminus t_g} ((\frac{1}{8m}(8|A^O_{i,k}| + 4|B^O_{i,k}| + 4|C^O_{i,k}|) + \frac{1}{2m}(2|A_{i,k} \setminus A^O_{i,k}| + 2|D_{i,k} \setminus D^O_{i,k}|))$$

Merging the terms:

$$= \sum_{t_g} \sum_{v, i\in E_{i,t_g}} |\alpha_{vi}|^2 \frac{1}{s} \prod_{k=\{1,\ldots,s\}\setminus t_g} \frac{1}{2m}(2|A_{i,k}| + |B^O_{i,k}| + |C^O_{i,k}| + 2|D_{i,k} \setminus D^O_{i,k}|)$$

Using the fact that for every k, $|A_{i,k}| + |B_{i,k}| + |C_{i,k}| + |D_{i,k}| = m$:

$$\le \sum_{t_g} \sum_{v, i\in E_{i,t_g}} |\alpha_{vi}|^2 \frac{1}{s} \prod_{k=\{1,\ldots,s\}\setminus t_g} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|)$$

The conditions $i \in E_{i,t_g}$ that we obtained at the first part of the proof are translated to $|B_{i,t_g}| + |C_{i,t_g}| + |D^O_{i,t_g}| \geq 1$. In order to be able to use these conditions we need to rewrite the formula. First we expand it:

$$= \frac{1}{s}\Big(\sum_{v,i\in E_{i,1}} |\alpha_{vi}|^2 \prod_{k=\{2,3,\ldots,s\}} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|)$$

$$+ \sum_{v,i\in E_{i,2}} |\alpha_{vi}|^2 \prod_{k=\{1,3,4,\ldots,s\}} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|)$$

$$\ldots + \sum_{v,i\in E_{i,d}} |\alpha_{vi}|^2 \prod_{k=\{1,2,\ldots,s-1\}} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|)\Big)$$

We denote the product term $\prod_{k=\{1,2,3,\ldots,s\}\setminus z} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|)$ as $P_{i,z}$. We also denote each set $\{E^*_{i,1} \cap E^*_{i,2} \cap \ldots \cap E^*_{i,s}\}$, where each term $E^*_{i,w}$ is either $E_{i,w}$ or its complement, $E^C_{i,w}$, depending on whether the w-th value of a binary vector $\boldsymbol{y}$ (size s) is 1 or 0 respectively, as $W_{i,\boldsymbol{y}}$. Then we have:

$$= \frac{1}{s}\Big(\sum_{\boldsymbol{y}\setminus(0\ldots0)} \sum_{i\in W_{i,\boldsymbol{y}},v} (|\alpha_{vi}|^2 \sum_{\{z:y_z=1\}} P_{i,z})\Big)$$

Let the function $\#\boldsymbol{y}$ give the number of positions i such that y_i=1.

$$= \frac{1}{s}\Big(\sum_{k=1}^{s} \sum_{\{\boldsymbol{y}:\#\boldsymbol{y}=k\}} \sum_{i\in W_{i,\boldsymbol{y}},v} (|\alpha_{vi}|^2 \sum_{\{z:y_z=1\}} P_{i,z})\Big)$$

We separately consider the following term for any arbitrary $\boldsymbol{y}$ with $\#\boldsymbol{y} = r$.

$$\sum_{i\in W_{i,\boldsymbol{y}}} (|\alpha_{vi}|^2 \sum_{\{z:y_z=1\}} P_{i,z})$$

The condition $i \in W_{i,\boldsymbol{y}}$ means that the following conditions hold together: $\{|B_{i,w}| + |C_{i,w}| + |D^O_{i,w}| \geq 1 : y_w = 1\}$,$\{|B_{i,w}| + |C_{i,w}| + |D^O_{i,w}| = 0 : y_w = 0\}$. We expand:

$$= \sum_{i\in W_{i,\boldsymbol{y}}} (|\alpha_{vi}|^2 \sum_{\{z:y_z=1\}} \prod_{k=\{1,2,3,\ldots,s\}\setminus z} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|)$$

$$= \sum_{i\in W_{i,\boldsymbol{y}}} (|\alpha_{vi}|^2 \sum_{\{z:y_z=1\}} \prod_{\{k:y_k=1,k\neq z\}} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|)$$

$$\prod_{\{k:y_k=0\}} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|)$$

And by using the above conditions:

$$\leq \sum_{i\in W_{i,\boldsymbol{y}}} (|\alpha_{vi}|^2 \sum_{\{z:y_z=1\}} \prod_{\{k:y_k=1,k\neq z\}} \frac{1}{2m}(2m-1) \prod_{\{k:y_k=0\}} \frac{1}{2m}(2m)$$

$$= \sum_{i\in W_{i,\boldsymbol{y}}} (|\alpha_{vi}|^2 \sum_{\{z:y_z=1\}} \left(\frac{2m-1}{2m}\right)^{r-1}$$

$$= \sum_{i\in W_{i,\boldsymbol{y}}} |\alpha_{vi}|^2 r \left(\frac{2m-1}{2m}\right)^{r-1}$$

Therefore the bound of our failure probability will be:

$$\begin{aligned} p_{\text{incorrect}} &\leq \frac{1}{s}(\sum_{k=1}^{s} \sum_{\{\boldsymbol{y}:\#\boldsymbol{y}=k\}} \sum_{i\in W_{i,\boldsymbol{y}},v} |\alpha_{vi}|^2 k \left(\frac{2m-1}{2m}\right)^{k-1}) \\ &= \frac{1}{s}(\sum_{k=1}^{s} k \left(\frac{2m-1}{2m}\right)^{k-1} \sum_{\{\boldsymbol{y}:\#\boldsymbol{y}=k\}} \sum_{i\in W_{i,\boldsymbol{y}},v} |\alpha_{vi}|^2) \\ &= \frac{1}{s}(\sum_{k=1}^{s} c_k k \left(\frac{2m-1}{2m}\right)^{k-1}) \end{aligned}$$

where $c_k = \sum_{\{\boldsymbol{y}:\#\boldsymbol{y}=k\}} \sum_{i\in W_{i,\boldsymbol{y}},v} |\alpha_{vi}|^2$
subject to conditions:

$$\sum_{k=1}^{s} c_k \leq 1 \tag{15}$$

and

$$\forall k : c_k \geq 0 \tag{16}$$

◀

Proof of Lemma 7. First we define state $|q_i\rangle$ as:

$$\begin{aligned} &i \in D && |q_i\rangle \equiv |d_i\rangle \\ &i \notin D && |q_i\rangle \equiv (\prod_{\{j:j\sim i, j\in D\}} Z^{d_j})|+_{\theta_i}\rangle \end{aligned}$$

By substituting $\rho_{M_k^\nu}$'s and taking the trace over R:
If $k \neq t_g$ the state becomes:

$$\sum_{\nu_{C,k}} p(\nu_{C,k})(\bigotimes_{i=km-n+1}^{km} |q_i\rangle\langle q_i| \bigotimes_{i=(k-1)m+n+1}^{km-n} \left(\left|\delta_i^{\boldsymbol{b'},\nu}\right\rangle\left\langle\delta_i^{\boldsymbol{b'},\nu}\right| \otimes |q_i^\nu\rangle\langle q_i^\nu|\right) \bigotimes_{i=1}^{2} \left(\left|\delta_{p_{i,k}}^{\boldsymbol{b'},\nu}\right\rangle\left\langle\delta_{p_{i,k}}^{\boldsymbol{b'},\nu}\right| \otimes \left|q_{p_{i,k}}^\nu\right\rangle\left\langle q_{p_{i,k}}^\nu\right|\right) \otimes I_{4(n-2)}/2^{4(n-2)})$$

where $\left|q_{p_{i,k}}^\nu\right\rangle$ denote the first layer pure qubits (a maximum of two) of the k-th graph state, used as padding (dummies) or trap and their positions are defined as: $1 + (k-1)m \leq \{p_{1,k}, p_{2,k}\} \leq n + (k-1)m$.

Otherwise, if $k = t_g$ the state becomes:

$$\sum_{\nu_{C,k}} p(\nu_{C,k})(\bigotimes_{i=t_g m-n+1}^{t_g m} |q_i\rangle\langle q_i| \bigotimes_{i=(t_g-1)m+n+1}^{t_g m-n} \left(\left|\delta_i^{\boldsymbol{b'},\nu}\right\rangle\left\langle\delta_i^{\boldsymbol{b'},\nu}\right| \otimes |q_i^\nu\rangle\langle q_i^\nu|\right) \otimes \left|\delta_u^{\theta_u,r_u}\right\rangle\left\langle\delta_u^{\theta_u,r_u}\right| \otimes \left|q_u^{\theta_u}\right\rangle\left\langle q_u^{\theta_u}\right| \otimes I_{4(w-1)}/2^{4(w-1)})$$

where $u = (t_g-1)m+1$ is the position of the single pure qubit of the input to the DQC1-MBQC computation.

An implicit assumption was that all δ's that are used to implement the measurements of maximally mixed inputs are maximally mixed states themselves, without any loss of generality.

We define a new controlled unitary:

$$\mathcal{P'}_k = \left(\prod_{\{i:i\notin D,(k-1)m+1\leq i\leq km-n\}} Z_i(-\delta_i) \right) \prod_{\{i:i\notin D_k\}} \prod_{\{j:j\sim i, j\in D_k\}} Z_i(d_j) \tag{17}$$

where D_k denotes the set of dummies of subgraph G'_k.

Using this unitary we rewrite the state. If $k \neq t_g$ it becomes:

$$\sum_{\nu_{C_k}} p(\nu_{C,k}) \mathcal{P'}^{\dagger} \mathcal{P'} (\bigotimes_{i=km-n+1}^{km} |q_i\rangle\langle q_i| \bigotimes_{i=(k-1)m+n+1}^{km-n} \left(\left|\delta_i^{\mathbf{b'},\nu}\right\rangle\left\langle\delta_i^{\mathbf{b'},\nu}\right| \otimes |q_i^{\nu}\rangle\langle q_i^{\nu}| \right) \bigotimes_{i=1}^{2} \left(\left|\delta_{p_{i,k}}^{\mathbf{b'},\nu}\right\rangle\left\langle\delta_{p_{i,k}}^{\mathbf{b'},\nu}\right| \otimes \left|q_{p_{i,k}}^{\nu}\right\rangle\left\langle q_{p_{i,k}}^{\nu}\right| \right) \otimes I_{4(n-2)}/2^{4(n-2)}) \mathcal{P'}^{\dagger} \mathcal{P'}$$

Otherwise:

$$\sum_{\nu_{C,k}} p(\nu_{C,k}) \mathcal{P'}^{\dagger} \mathcal{P'} \bigotimes_{i=t_g m-n+1}^{t_g m} |q_i\rangle\langle q_i| \bigotimes_{i=(t_g-1)m+n+1}^{t_g m-n} \left(\left|\delta_i^{\mathbf{b'},\nu}\right\rangle\left\langle\delta_i^{\mathbf{b'},\nu}\right| \otimes |q_i^{\nu}\rangle\langle q_i^{\nu}| \right) \otimes \left|\delta_u^{\theta_u,r_u}\right\rangle\left\langle\delta_u^{\theta_u,r_u}\right| \otimes \left|q_u^{\theta_u}\right\rangle\left\langle q_u^{\theta_u}\right| \otimes I_{4(w-1)}/2^{4(w-1)}) \mathcal{P'}^{\dagger} \mathcal{P'}$$

After applying the innermost unitary, if $k \neq t_g$:

$$\sum_{\nu_{C,k}} p(\nu_{C,k}) \mathcal{P'}^{\dagger} (\bigotimes_{i=km-n+1}^{km} |q'_i\rangle\langle q'_i| \bigotimes_{i=(k-1)m+n+1}^{km-n} \left(\left|\delta_i^{\mathbf{b'},\nu}\right\rangle\left\langle\delta_i^{\mathbf{b'},\nu}\right| \otimes \left|q_i^{'\nu}\right\rangle\left\langle q_i^{'\nu}\right| \right) \bigotimes_{i=1}^{2} \left(\left|\delta_{p_{i,k}}^{\mathbf{b'},\nu}\right\rangle\left\langle\delta_{p_{i,k}}^{\mathbf{b'},\nu}\right| \otimes \left|q_{p_{i,k}}^{'\nu}\right\rangle\left\langle q_{p_{i,k}}^{'\nu}\right| \right) \otimes I_{4(n-2)}/2^{4(n-2)}) \mathcal{P'}$$

where state $|q'_i\rangle$ is defined as:

$$\begin{aligned} i \in D \qquad & |q'_i\rangle \equiv |d_i\rangle \\ i \notin D, \forall k : km \geq i \geq km-n+1 \qquad & |q'_i\rangle \equiv |+_{\theta_i}\rangle \\ i \notin D, \forall k : km-n \geq i \geq (k-1)m+1 \qquad & |q'_i\rangle \equiv \left| +_{-a_i'' \, \mathbf{b'},\mathbf{r}<i \, -r_i\pi} \right\rangle \end{aligned}$$

Otherwise, if $k = t_g$:

$$\sum_{\nu_{C,k}} p(\nu_{C,k}) \mathcal{P'}^{\dagger} (\bigotimes_{i=t_g m-n+1}^{t_g m} |q'_i\rangle\langle q'_i| \bigotimes_{i=(t_g-1)m+n+1}^{t_g m-n} \left(\left|\delta_i^{\mathbf{b'},\nu}\right\rangle\left\langle\delta_i^{\mathbf{b'},\nu}\right| \otimes \left|q_i^{'\nu}\right\rangle\left\langle q_i^{'\nu}\right| \right) \otimes \left|\delta_u^{\theta_u,r_u}\right\rangle\left\langle\delta_u^{\theta_u,r_u}\right| \otimes \left|q_u^{'\theta_u}\right\rangle\left\langle q_u^{'\theta_u}\right| \otimes I_{4(w-1)}/2^{4(w-1)}) \mathcal{P'}$$

It is essential for the proof that each term with index i in the tensor product depends only on parameters with index $\leq i$ and the term with index $(t_g-1)m+1$ (input qubit) and

the trap qubit and its measurement angle (if it is not an output) depend only on their own parameters. This allows to break the summations and calculate them iteratively from left to right, given the following:

$$\sum_{d_i} p(d_i)|d_i\rangle\langle d_i| = \frac{I}{2}$$

$$\sum_{\theta_i} p(\theta_i)|+_{\theta_i}\rangle\langle+_{\theta_i}| = \frac{I}{2}$$

$$\sum_{\theta_i,r_i,d_i} p(\theta_i,r_i,d_i)\left|\delta_i^{\mathbf{b'},\nu}\right\rangle\left\langle\delta_i^{\mathbf{b'},\nu}\right| \otimes |d_i\rangle\langle d_i| = \frac{I_4}{2^4}$$

$$\begin{aligned}
&\sum_{\theta_i,r_i} p(\theta_i,r_i)\left|\delta_i^{\mathbf{b'},\nu}\right\rangle\left\langle\delta_i^{\mathbf{b'},\nu}\right| \otimes \left|+_{-a_i''^{\,\mathbf{b'},\mathbf{r}<i}-r_i\pi}\right\rangle\left\langle+_{-a_i''^{\,\mathbf{b'},\mathbf{r}<i}-r_i\pi}\right| \\
&= \sum_{r_i} p(r_i)\left(\sum_{\theta_i} p(\theta_i)\left|a_i''^{\,\mathbf{b'},\mathbf{r}<i}+\theta_i+r_i\pi\right\rangle\left\langle a_i''^{\,\mathbf{b'},\mathbf{r}<i}+\theta_i+r_i\pi\right|\right) \\
&\qquad\otimes\left|+_{-a_i''^{\,\mathbf{b'},\mathbf{r}<i}-r_i\pi}\right\rangle\left\langle+_{-a_i''^{\,\mathbf{b'},\mathbf{r}<i}-r_i\pi}\right| \\
&= \sum_{r_i} p(r_i)\frac{I_3}{2^3} \otimes \left|+_{-a_i''^{\,\mathbf{b'},\mathbf{r}<i}-r_i\pi}\right\rangle\left\langle+_{-a_i''^{\,\mathbf{b'},\mathbf{r}<i}-r_i\pi}\right| \\
&= \frac{I_4}{2^4}
\end{aligned}$$

where $I_n = \bigotimes_n I$. The last step was possible because each corrected computation angle a_i'' depends only on past r's.

And finally (for $u = (t_g-1)m+1$),

$$\begin{aligned}
&\sum_{\theta_u,r_u} p(\theta_u,r_u)\left|\delta_u^{\theta_u,r_u}\right\rangle\left\langle\delta_u^{\theta_u,r_u}\right| \otimes \left|+_{-a_u'-r_u\pi}\right\rangle\left\langle+_{-a_u'-r_u\pi}\right| \\
&= \sum_{r_u} p(r_u)\left(\sum_{\theta_u} p(\theta_u)\left|a_u'+\theta_u+r_u\pi\right\rangle\left\langle a_i'+\theta_u+r_u\pi\right|\right) \\
&\qquad\otimes\left|+_{-a_u'-r_u\pi}\right\rangle\left\langle+_{-a_u'-r_u\pi}\right| \\
&= \frac{I_4}{2^4}
\end{aligned}$$

For $k \neq t_g$, if $km \geq t_k \geq km-n+1$ the above procedure will eventually give:

$$\begin{aligned}
&\mathcal{P}'^{\dagger}\left(\frac{I_{4m-3n-1}}{2^{4m-3n-1}} \otimes \left|+_{\theta_{t_k}}\right\rangle\left\langle+_{\theta_{t_k}}\right|\right)\mathcal{P}' \\
&= \frac{I_{4m-3n-1}}{2^{4m-3n-1}} \otimes \left|+_{\theta_{t_k}}\right\rangle\left\langle+_{\theta_{t_k}}\right|
\end{aligned}$$

If $km-n \geq t_k \geq (k-1)m+1$ the above procedure will eventually give:

$$\begin{aligned}
&\mathcal{P}'^{\dagger}\left(\frac{I_{4m-3n-4}}{2^{4m-3n-4}} \otimes \left|\delta_{t_k}^{\nu_T}\right\rangle\left\langle\delta_{t_k}^{\nu_T}\right| \otimes \left|+_{r_{t_k}\pi}\right\rangle\left\langle+_{r_{t_k}\pi}\right|\right)\mathcal{P}' \\
&= \frac{I_{4m-3n-4}}{2^{4m-3n-4}} \otimes \left|\delta_{t_k}^{\nu_T}\right\rangle\left\langle\delta_{t_k}^{\nu_T}\right| \otimes \left|+_{\theta_{t_k}}\right\rangle\left\langle+_{\theta_{t_k}}\right|
\end{aligned}$$

And for $k = t_g$ the result will be: $\bigotimes_{4m-3n} I$, which concludes the proof. ◀